CONTENTS

BIOINFORMATICS

Vol. 22 no 14 2006, pages e1–e2
doi:10.1093/bioinformatics/btl373

Editorial

ISMB 2006

This volume contains the papers accepted for presentation at the 2006 Intelligent Systems for Molecular Biology conference (ISMB 2006; www.iscb.org/ismb2006) held in Fortaleza, Brazil from August 6-10, 2006. The conference is the annual meeting of the International Society for Computational Biology (ISCB). The papers presented here are noteworthy for several reasons. First, papers are open access and freely available to the worldwide community ahead of the conference and subsequently form an on-line only issue of the journal Bioinformatics. Second, the review process was conducted slightly differently to previous years. Finally, and most importantly, we believe that the scientific content to be outstanding. This indicates both the strength of the field and a desire to present the best work in an important part of the world for the first time, making a truly international meeting and ISCB a truly international society.

The call for papers resulted in 404 submissions in one of thirteen different categories (Table 1). Area Chairs were recruited for each category and they in turn assigned reviewers for each paper. All papers received two or three reviews. Based on the reviews Area Chairs made recommendations on the papers to be presented. The final program was then decided by the Program Chairs based on these recommendations and the need to provide a balanced program. As a result 67 papers are included making for an acceptance rate of 16.6%.

A total of 347 papers had authors from a single country, 52 from 2 countries, 4 from 3 countries and 1 from 4 countries, respectively. Of the 347 papers from one country, 180 were from North America, 90 were from Europe and Israel, 49 from Asia, 16 from South America, 11 from Australia and 1 from Africa.

The papers accepted (Table 1) were organized slightly differently from previous years, with a category for Human Health added. This resulted in 20 submissions indicating strong interest from the community. Conversely, molecular and supramolecular dynamics was added and only resulted

Table 1. ISMB 2006 Program Areas, Program Chairs and Paper Distribution

Area	Area Chairs	Papers Received/Accepted
Comparative Genomics	Koonin, Eugene Claverie, Jean-Michel	25/4
Databases & Data Integration	Kanehisa, Minoru Apweiler, Rolf	31/3
Evolution and Phylogeny	Gouy, Manolo Warnow, Tandy	21/3
Human Health	Ofran, Yanay Radivojac, Predrag Kann, Maricel Punta, Marco	20/3
Molecular and Supramolecular Dynamics	Murray, Diana Shakhnovich, Eugene	5/1
Ontologies	Kumar, Anand Stevens, Robert Parkinson, Helen	8/2
Proteomics	Zhang, Zhaolei Troyanskaya, Olga	38/6
Sequence Analysis	Grishin, Nick Brunak, Soren	59/10
Structural Bioinformatics	Thornton, Janet Russell, Rob	51/8
Systems Biology	Bader, Joel Sander, Chris	55/11
Text Mining & Information Extraction	Valencia, Alfonso Rzhetsky, Andrey	33/6
Transcriptomics	Margalit, Hanah Zhang, Michael	42/7
Miscellaneous	Bourne, Philip	16/3

in 5 submissions. Other areas remained approximately proportional to 2005 submissions.

We would like to thank the Area Chairs and reviewers for the quality they have brought to the conference, to Richard van de Stadt for support with Cyberchair in managing the selection process and Steven Leard for following up with many of the loose ends needed to produce the volume in a timely manner.

Philip E. Bourne and Soren Brunak, Program Chairs
ISMB 2006

BIOINFORMATICS

Vol. 22 no 14 2006, pages e3–e7
doi:10.1093/bioinformatics/btl374

ISMB 2006 Organization

CONFERENCE CHAIR

Goran Neshich, EMBRAPA/CNPTIA, Campinas, Brazil

CONFERENCE VICE-CHAIR

Ana Tereza Ribeiro de Vasconcelos, LNCC/MCT, Petrópolis, Brazil

STEERING COMMITTEE

Barb Bryant, ISCB Vice President, Millennium Pharmaceuticals, Cambridge, MA, USA

Michael Gribskov, ISCB President, Purdue University, West Lafayette, IN, USA

Janet Kelso, ISCB Conferences Committee Chair, Max-Planck-Institute for Evolutionary Anthropology, Leipzig, Germany

Steven Leard, ISCB/ISMB Conference Liaison, Edmonton, AB, Canada

BJ Morrison McKay, ISCB Executive Officer, San Diego, CA, USA

Goran Neshich, ISMB 2006 Conference Chair, EMBRAPA/CNPTIA, Campinas, Brazil

David Rocke, Steering Committee Chair/ISCB Treasurer, University of California, Davis, CA, USA

Hershel Safer, ISCB Conferences Committee Past Chair, Weizmann Institute of Science, Rehovot, Israel

David States, ISMB 2005 Conference Chair, University of Michigan, Ann Arbor, MI, USA

SCIENTIFIC ORGANIZING COMMITTEE

Amos Bairoch, Conference Advisor (Europe), SwissProt, Geneva, Switzerland

Junior Barreira, Poster Session Chair, USP, São Paulo-SP

Phil Bourne, Program Committee Co-Chair, University of California, San Diego, USA

Søren Brunak, Program Committee Co-Chair, University of Denmark, Lyngby, Denmark

Barbara Bryant, PLoS Track Oral Presentations Chair, Millennium Pharmaceuticals, Cambridge, MA, USA

Ana Tereza Ribeiro de Vasconcelos, Vice-Chair, LNCC/MCT, Petrópolis, Brazil

Nobuhiro Go, Conference Advisor (Asia-Australia), Jaeri, Kyoto, Japan

Barry Honig, Conference Advisor (North America), Columbia University, New York, USA

Goran Neshich, Chair, EMBRAPA/CNPTIA, Campinas, Brazil

Shoba Ranganathan, Tutorial Session Chair, Macquarie University, Sydney, Australia

Sandro Souza, Scientific Demo Chair, Ludwig Inst., São Paulo-SP

David States, Past Conference Program Chair, University of Michigan, Ann Arbor, MI, USA

PROGRAM COMMITTEE CHAIRS

Phil Bourne, University of California, San Diego, USA

Søren Brunak, University of Denmark, Lyngby, Denmark

AREA CHAIRS

Comparative Genomics

Jean-Michel Claverie, University of Mediterranée, Marseilles, France

Eugene Koonin, National Center for Biotechnology Information (NCBI), National

Library of Medicine (NLM), National Institutes of Health (NIH), Bethesda, MD, USA

Databases & Data Integration

Rolf Apweiler, European Bioinformatics Institute, Cambridge, UK

Minoru Kanehisa, Kyoto University, Kyoto, Japan

Evolution and phylogeny

Manolo Gouy, Université Claude Bernard, Lyon, France

Tandy Warnow, The University of Texas, Austin, USA

Human Health

Maricel Kann, National Center for Biotechnology Information, NIH., Rockville, MD, USA

Marco Punta, Columbia University, New York, USA

Predrag Radivojac, University of Indiana, Bloomington, IN, USA

Yanay Ofran, Columbia University, New York, USA

Molecular and Supramolecular Dynamics

Diana Murray, Cornell University, Ithaca, NY, USA

Eugene Shakhnovich, Harvard University, Cambridge, MA, USA

Ontologies

Anand Kumar, Centre for Mathematical Modelling and Computer Simulation, Bangalore, India

Helen Parkinson, European Bioinformatics Institute, Cambridge, UK

Robert Stevens, The University of Manchester, UK

Proteomics

Olga Troyanskaya, Princeton University, Princeton, NJ, USA

Zhaolei Zhang, University of Toronto, Canada

Sequence Analysis

Søren Brunak, University of Denmark, Lyngby, Denmark

Nick Grishin, University of Texas Southwestern Medical Center, Dallas, USA

Structural Bioinformatics

Rob Russell, European Molecular Biology Laboratory, Heidelberg, Germany

Janet Thornton, European Bioinformatics Institute, Cambridge, UK

Systems Biology

Joel Bader, Johns Hopkins University, Baltimore, MD, USA

Chris Sander, Memorial Sloan-Kettering Cancer Center, New York, USA

Text Mining & Information Extraction

Andrey Rzhetsky, Columbia University, New York, USA

Alfonso Valencia, Centro Nacional de Biotecnologia, Madrid, Spain

Transcriptomics

Hanah Margalit, The Hebrew University of Jerusalem, Israel

Michael Zhang, Cold Spring Harbor Laboratory, NY, USA

PROGRAM COMMITTEE MEMBERS

John Aach
Eugene Agichtein
Uri Akavia
Tatsuya Akutsu
Sophia Ananiadou
Miguel Andrade
Justen Andrews
Michael Ashburner
Francisco Azuaje
Mike Bada
Gary Bader
David Bader
Joel Bader
Timothy Bailey
Pierre Baldi
Donald Bashford
Kalyan Basu
Alex Bateman
Andy Baxevanis
Ronald Beavis
David Beer
Asa Ben-Hur
Bonnie Berger
Helen Berman
Doron Betel
Matthew Betts
David Beveridge
Jadwiga Bienkowska
Henry Bigelow
Ewan Birney
Judith Blake
Eric Blalock
Christian Blaschke
Olivier Bodenreider
Anthony Bonner
Alexandre Bonvin
Philip Bourne
Paul Brazhnik
Alvis Brazma
Kristine Briedis
Yana Bromberg
Bernie Brooks
Steve Bryant
Philipp Bucher
Jeremy Buhler
Martha Bulyk
Peter Buneman
Harmen Bussemaker
Liming Cai
Andrea Califano
Sergi Castellano
Gianni Cesareni
Nitesh Chawla
Thomas Cheatham
Jake Chen
Ting Chen
Kei Cheung
Jo-Lan Chung
Kevin Cohen
Sarah Cohen-Boulakia
Nigel Collier
Jim Collins
Markus Covert
Mehmet Dalkilic
Vlado Dancik
Gautam Dantas
Rajat De
Jaime de la Rocha
Ulrik de Lichtenberg
Marco De Vivo
Charlotte Deane
Eric Deeds
Michael Deem
Emek Demir
Chris Dobson
Stijn van Dongen
Chris Dupont
William Eaton
Robert Edgar
Ingvar Eidhammer
Michael Eisen
Frank Eisenhaber
Arne Elofsson
Andrew Emili
Anton Enright
Fazel Famili
Piero Fariselli
Howard Feldman
Sarel Flieshman
Chris Floudas
Federico Fogolari
Ernest Fraenkel
Iddo Friedberg

Carol Friedman
Nir Friedman
Dmitrij Frishman
Ken Fukuda
Ora Furman
Stefano Fusi
Terry Gaasterland
Nicolas Galtier
Jiali Gao
Angel Garcia
Paul Gardner
Tim Gardner
Olivier Gascuel
Daniel Gautheret
Mikhail Gelfand
Mark Gerstein
Andy Gibson
David Gifford
Alejandro Giorgetti
Galina Glazko
Carole Goble
Adam Godzik
Takashi Gojobori
Nick Goldman
Richard Goldstein
Anders Gorm Pedersen
Jan Gorodkin
Andrew Goryachev
Julian Gough
John Goutsias
Manolo Gouy
Apostol Gramada
Jenny Gu
Katia Guimaraes
Dan Gusfield
Matthew Hahn
Udo Hahn
Sridhar Hannenhalli
Midori Harris
Alex Hartemink
Vasileios Hatzivassiloglou
Kim Henrick
Bernard Henrissat
Jaap Heringa
Henning Hermjakob
Hanspeter Herzel
Des Higgins
Lynette Hirschman
Robert Hoffmann
Mark Holder
Liisa Holm
Ian Holmes
Evig Hovig
Earl Hubbel
Larry Hunter
Lilia Iakoucheva
Trey Ideker
Tommi Jaakkola
Lars Jermiin
Tao Jiang
Inge Jonassen
Ziv Bar Joseph
Cliff Joslyn
Tamer Kahveci
Vipul Kashyap
Scott Kelley
Manolis Kellis
Graham Kemp
Andrew Kernytsky
Paul Kersey
Lars Kiemer
Sun Kim
Roy Kishony
Steven Kleinstein
Edda Klipp
Isaac Kohane
Eugene Kolker
Eugene Koonin
Martin Krallinger
Roland Krause
Michael Krauthammer
Gabriel Kreiman
Jan Kubelka
Jens Lagergren
Nicolas Lartillot
Roman Laskowski
Doheon Lee
Christopher Lee
Jim Leebens-Mack
Pierre Legrain
Christina Leslie
Stan Letovsky
Hao Li
Ming Li
Hao Li
Randy Linder
Stinus Lindgreen
Jian Liu
XiaoleLiu
Phil Lord
Joanne Luciano
Bin Ma
Bob MacCallum
Michael MacCoss
Tom Madej
William Majoros
Cristina Marchetti
Leonardo Marino
Stanke Mario
Victor Markowitz
Troels Marstrand
Robin McEntire
Liam McGuffin
Joe Mellor
Fan Meng
Hans-Werner Mewes
Irmtraud Meyer
Ivana Mihalek
Madan Mohan
Bernard Moret
Steve Mount
Parvin Mousavi
Nicola Mulder
Kasper Munch
Diana Murray
Arcady Mushegian
Jose Nacher
Luay Nakhleh
Victor Neduva
Sven Nelander
Goran Nenadic
Henrik Nielsen
Kang Ning
Bill Noble
Cedric Notredame
Ruth Nussinov
Uwe Ohler
Tom Oldfield
Christine Orengo
Christos Ouzounis
Anna Panchenko
Vijay Pande
Kangs Pandjassarame
Jong Park
Andrea Passerini
Yalini Pathy
Linda Pattini
Paul Pavlidis
Florencio Pazos
Dana Peer
Matteo Pellegrini
Guy Perriere
Mihaela Pertea
Pavel Pevzner
Yizhak Pilpel
Ron Pinter
Gianluca Pollastri
Julia Ponomarenko
Amol Prakash
Natasa Przulj
Dariusz Przybylski
Teresa Przytycka
Tal Pupko
John Quackenbush
Shalom Rackovsky
Jagath Rajapakse
Nikolaus Rajewsky
Sanjay Ranka

Magnus Rattray
Dietrich Rebholz-Schuhmann
Alan Rector
Aviv Regev
Marcel Reinders
Boris Reva
Vicente Reyes
John Rice
Isidore Rigoutsos
David Rocke
Allen Rodrigo
Fritz Roth
Frederic Rousseau
Pierre Rouze
Eitan Rubin
Patrick Ruch
Eytan Ruppin
Alan Ruttenberg
Andrey Rzhetsky
Lukasz Salwinski
Chris Sander
Clare Sansom
Akinori Sarai
Eric Scheeff
Avner Schlessinger
Michael Schrorder
Klaus Schulten
Steffen Schulze-Kremer
Torsten Schwede
Jonathan Sebat
Ng See-Kiong
Eran Segal
Bo Servenius
Boris Shakhnovich
Ron Shamir
Roded Sharan
HagitShatkay
Ilya Shindyalov
Ilya Shmulevich
Adam Siepel
Mona Singh
Ambuj Singh
Steve Skiena
Barry Smith
Nick Socci
Johannes Soeding
Victor Solovyev
Terry Speed
Paul Spellman
John Spouge
Alexander Stark
David States
Mike Steel
Gustavo Stolovitzky
James Stroud
Fengzhu Sun
Sean Sun
Wing Sung
Fariza Tahi
Michio Takahashi
Lorrie Tanabe
Amos Tanay
Haixu Tang
Yoshio Tateno
Willie Taylor
Michael Thompson
Annabel Todd
Anna Tramontano
Gabor Tusnady
Alfonso Valencia
Lisa Vawter
Stella Veretnik
Karin Vespoor
Anne-Lise Veuthey
Jaak Vilo
Bafna Vineet
Martin Vingron
Dennis Vitkup
Yury Vorobjev
Slobodan Vucetic
Alessandro Vullo
Dennis Wall
Tandy Warnow
Wyeth Wasserman
Bertram Weiss
Zhiping Weng
John Westbrook
David Westhead
Daniela Wieser
Chris Wiggins
John Wilbur
Jennifer Williams
Edgar Wingender
Shoshana Wodak
Yuri Wolf
Limsoon Wong
Jonathan Wren
Ioannis Xenarios
Lei Xie
Eric Xing
Dong Xu
Ying Xu
Zohar Yakhini
Akihiko Yamagishi
Dan Yamins
Bo Yan
Song Yang
Yuzhen Ye
Haiyuan Yu
Krystyna Zakrzewska
Michael Zhang
Q Cindy Zhang
Hongyu Zhao
Xianghong Zhou
Ralf Zimmer
Igor Zwir

PLOS TRACK CO-CHAIRS

Barb Bryant, Chair, Millennium Pharmaceuticals, Cambridge, MA, USA

Junior Barrera, Co-Chair, USP, São Paulo-SP, Brazil

Phil Bourne, Co-Chair, University of California, San Diego, USA

Steven Brenner, Co-Chair, Unviversity of California, Berkley, USA

PLOS TRACK COMMITTEE MEMBERS

Adam Arkin, University of California, Berkeley, USA

Sandro José de Souza, Ludwig Inst., São Paulo-SP, Brazil

Lyle Graham, Laboratoire de Neurophysique et Physiologie du Système Moteur, Paris, France

Lawrence Hunter, University of Colorado Health Sciences Center, Aurora, CO, USA

Michal Linial, The Hebrew University of Jerusalem, Jerusalem, Israel

Burkhard Rost, Columbia University, New York, USA

Andrey Rzhetsky, Columbia University, New York, USA

POSTER SESSION CHAIRS

Junior Barrera, USP, São Paulo-SP, Brazil

Fernando Barroso, USP, Riberão Preto-SP, Brazil

POSTERS COMMITTEE MEMBERS

Hugo A. Armelin
Junior Barrera
Paulo M. Bisch
Sandro Luis Bonatto
Helena Paula Brentani
Marcelo Briones
Helaine Carrer
Luciano F. Costa
Fernando Luis Barroso da Silva
Flávio Soares Correa da Silva
Antonio Francisco de Araujo
Hernando del Partilho
Chuck Farah
João Eduardo Ferreira
Douglas S. Galvão
Silvana Giuliatti
Gustavo H. Goldman
Ronaldo Fumio Hashimoto
João Paulo Kitajima
Marcos Antonio Machado
Rogerio Meneghini
Diogo Meyer
Maria Carolina Monard
Ulisses Braga Neto
Glaucius Oliva
Pedro G. Pascutti
Luiz Fernando Lima Reis
João Setubal
Munir S. Skaf
Caetano Traina
Renata Wassermann
Rita Zorzenon

LOCAL ORGANIZING COMMITTEE

Junior Barrera, USP, São Paulo-SP, Brazil

Fernando Barroso, USP, Riberão Preto-SP, Brazil

Diana Magalhães de Oliveira, UECE, Fortaleza, Brazil

Sandro José de Souza, Ludwig Inst., São Paulo-SP, Brazil

Paula Kuzer Falcão, EMBRAPA/CNPTIA, Campinas-SP, Brazil

Richard Charles Garrat, USP, São Carlos-SP, Brazil

João Paulo Kitajima, Alellyx Applied Genomics and LBI, Unicamp, Campinas-SP, Brazil

José Antonio Maranho, Vertical Events, Campinas-SP, Brazil

Goran Neshich, EMBRAPA/CNPTIA, Campinas-SP, Brazil

Álvaro Seixas Neto, EMBRAPA/CNPTIA, Campinas-SP, Brazil

Glacius Oliva, USP, São Carlos-SP, Brazil

Gonçalo A. G. Pereira, Unicamp, Campinas-SP, Brazil

João Carlos Setubal, Virginia Bioinformatics Institute, Virginia Tech, USA

Sergio Verjovski-Almeida, USP, São Paulo-SP, Brazil

LOGISTICAL ORGANIZING COMMITTEE

Fernando Barroso, USP, Riberão Preto-SP

João Kitajima, Alellyx Applied Genomics and LBI, Unicamp, Campinas-SP

Paula Kuser, EMBRAPA/CNPTIA, Campinas-SP

Steven Leard, ISCB/ISMB Conference Liaison, Edmonton, AB, Canada

Joe Maranho, Vertical Events, Campinas-SP, Brazil

Neusa Maranho, Vertical Events, Campinas-SP, Brazil

Goran Neshich, Conference Chair, EMBRAPA/CNPTIA, Campinas, Brazil

Douglas Umaki Morita, EMBRAPA/CNPTIA, Campinas, Brazil

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e9–e15
doi:10.1093/bioinformatics/btl213

Automatic clustering of orthologs and inparalogs shared by multiple proteomes

Andrey Alexeyenko[1,2], Ivica Tamas[1,3], Gang Liu[1] and Erik L.L. Sonnhammer[1,2,*]

[1]Center for Genomics and Bioinformatics, Karolinska Institutet, S-17177 Stockholm, [2]Stockholm Bioinformatics Center, Albanova, Stockholm University, SE-10691 Stockholm, Sweden and [3]Present address: Department of Molecular Biology & Functional Genomics, Stockholm University, SE-10691, Stockholm, Sweden

ABSTRACT

Motivation: The complete sequencing of many genomes has made it possible to identify orthologous genes descending from a common ancestor. However, reconstruction of evolutionary history over long time periods faces many challenges due to gene duplications and losses. Identification of orthologous groups shared by multiple proteomes therefore becomes a clustering problem in which an optimal compromise between conflicting evidences needs to be found.

Results: Here we present a new proteome-scale analysis program called MultiParanoid that can automatically find orthology relationships between proteins in multiple proteomes. The software is an extension of the InParanoid program that identifies orthologs and inparalogs in pairwise proteome comparisons. MultiParanoid applies a clustering algorithm to merge multiple pairwise ortholog groups from InParanoid into multi-species ortholog groups. To avoid outparalogs in the same cluster, MultiParanoid only combines species that share the same last ancestor.

To validate the clustering technique, we compared the results to a reference set obtained by manual phylogenetic analysis. We further compared the results to ortholog groups in KOGs and OrthoMCL, which revealed that MultiParanoid produces substantially fewer outparalogs than these resources.

Availability: MultiParanoid is a freely available standalone program that enables efficient orthology analysis much needed in the post-genomic era. A web-based service providing access to the original datasets, the resulting groups of orthologs, and the source code of the program can be found at http://multiparanoid.cgb.ki.se.

Contact: Erik.Sonnhammer@sbc.su.se

Supplementary information: http://multiparanoid.cgb.ki.se/ISMB2006/

1 INTRODUCTION

The increasing availability of complete proteomes provides the opportunity to reconstruct their evolutionary history based on sequence data. This is particularly welcomed by functional and comparative genomics, which is heavily dependent on orthology analysis. Orthologous genes exist in many guises, ranging from proteins with identical functions in identical pathways to proteins that share a common evolutionary origin but have diverged in function. Establishing orthology between genes is today one of the most reliable methods to obtain functional annotation.

In this paper we consider orthologs as defined by Fitch (1970): genes descending from a single gene in the last common ancestor of the species. Such genes are most likely to be functional counterparts. On the other hand, genes arising from duplications are defined as paralogs. Genomes of invertebrates and higher organisms are notorious for high numbers of gene duplications and/or gene losses. Such genomic variation has been explained as an adaptation to different environments (Chervitz *et al*., 1998; Troemel *et al*., 1995; Enmark and Gustafsson, 2001; Maglich *et al*., 2001).

Paralogs may arise from a duplication that occurred either before or after the speciation event that gave rise to the species of interest. If the duplication occurred first, the genes resulting from the duplication cannot be orthologs. Such genes are called outparalogs (Sonnhammer and Koonin, 2002). However, if the duplication happened after the speciation, the resulting genes can be considered co-orthologs. Such genes are called inparalogs. Given that the goal is to identify the complete set of orthologs and avoid non-orthologs, one wants to find all inparalogs while avoiding all outparalogs. A simplification of the problem would be to consider only the most similar inparalogs as true orthologs. However, there is often no clear functional distinction between inparalogs in the same group (Kondrashov *et al*., 2002).

The best orthology analysis is obtained from careful manual inspection of phylogenetic trees, for instance as was done by Wheelan *et al*., (1999) to identify human-mouse-rat-worm orthologs. However, this is very labor-intensive, and to save time many groups have resorted to using high-scoring global BLAST (Altschul *et al*., 1997) matches to approximate orthologs (e.g. Rubin *et al*., 2000). The BLAST approach can be substantially improved by only accepting reciprocally best matching protein pairs as orthologs (Mushegian *et al*., 1998). This approach works reasonably well for the proteomes of bacteria. However, its application to diversified eukaryotic species faces additional problems due to a complex evolutionary past (Xie and Ding, 2000).

The COG method (Tatusov *et al*., 1997) extends the reciprocal best matching method to allow incorporation of multiple species into each ortholog group. It has the ability to include inparalogs, but because it groups sequences of widely different evolutionary distances in a single cluster, out-paralogs are also commonplace. COGs

*To whom correspondence should be addressed.

initially contained only prokaryotic proteomes, but a version of seven eukaryotic species—KOGs—has been released (Tatusov *et al*., 2003). Lee *et al*. (2002) applied the COG method to cDNA sequences of 28 eukaryotes, resulting in the EGO (formerly TOGA) database.

OrthoMCL represents a different approach to finding multi-species ortholog groups. It uses a Markov clustering algorithm based on graph flow theory, and can find clusters of desired tightness depending on the ''inflation parameter'' (Li *et al*., 2003). With the parameters they used, OrthoMCL was much stricter than EGO and KOGs with regard to the inclusion of outparalogs. The OrthoMCL web resource initially included *E. coli* and nine eukaryotic proteomes; the latest release contains 55 proteomes (Chen *et al*., 2006). A drawback with the above methods is that they do not provide confidence values for the predicted orthologs. They also do not necessarily have a unique last common ancestor in each group, which can lead to inclusion of outparalogs in the same cluster.

The InParanoid method was specifically designed to find inparalogs by a special extension of the reciprocal best matching method in pairwise proteome comparisons (Remm *et al*., 2001). It provides confidence scores for both the seed orthologs and the inparalogs. The method was evaluated against a manually curated set of worm-human orthologous transmembrane proteins. The latest release of InParanoid contained 25 eukaryotic proteomes plus *E. coli* (O'Brien *et al*., 2005).

In this paper we employ a new clustering technique to keep the advantages of InParanoid while extending the method to include multiple species. The new method called MultiParanoid reads the output from InParanoid and builds multi-species clusters from these. To benchmark the method on three-species ortholog groups, we extended the manually curated reference dataset by also including fly orthologs.

We then used this curated dataset as a reference in order to estimate the quality and features of MultiParanoid. We also compared the results to KOGs and OrthoMCL, and carried out a detailed analysis of the differences. Each discrepancy was categorized to gain insights into the particular characteristics of each method. We also review the HomoloGene database (Wheeler *et al*., 2006) that was not directly comparable to MultiParanoid clusters.

2 METHODS

2.1 Algorithm

MultiParanoid takes pairwise ortholog clusters (from e.g. InParanoid) and merges them into multi-species clusters. While there is no formal limit on the number of proteomes that can be processed, the following description is given for the case of three species. The input to MultiParanoid for N species consists of $N * (N - 1) / 2$ tables of InParanoid output—one for each pair of species.

Given a list of species A, B, and C, and pairwise ortholog cluster tables A-B, B-C, and A-C, the procedure starts by reading the list of clusters from the A-B table. These are kept as seed clusters that may be extended to include sequences in the other proteomes. The program next looks for the presence of the seed orthologs from the A-B cluster in the A-C and B-C tables. If present, all the members (inparalogs) in corresponding A-C or B-C clusters are added to the seed cluster. This procedure is repeated until all pairwise ortholog groups are processed.

This clustering corresponds to a single-linkage approach. We also implemented additional cluster trimming features in order to exclude outliers. For instance every member was required to have the confidence value—an average of its InParanoid scores—above a cutoff. However, since InParanoid clusters are already strict, trimming the multi-species clusters did not improve the overall quality.

On rare occasions, a gene may be assigned to multiple MultiParanoid clusters. To address this problem, we applied an additional procedure to assure non-redundant presence of the analyzed genes in the clusters. If a gene is not a seed ortholog in any of the clusters, it is assigned to the cluster where it has a higher InParanoid score and removed from the other. If it is assigned as the seed ortholog of a cluster, it is retained in this cluster in order to avoid disrupting the processed cluster and deleted from the other.

2.2 Construction of the reference set

Clustering of worm proteins containing at least two transmembrane segments was originally done as described elsewhere (Remm and Sonnhammer, 2000). To retrieve homologous fly and human sequences, SWISS-PROT, TREMBL and VTS databases were searched using specifically designed HMMs. After a manual curation, the original dataset contained 221 group of proteins based on sequence similarities. The largest observed family consists mainly of G-protein coupled receptors.

Putative worm—fly—human orthologs were extracted via complete phylogenetic analysis as follows:

(i) Multiple sequence alignments were done with the HMMALIGN algorithm from the HMMER package (http://hmmer.wustl.edu). Sequences having gaps (>50%) were removed from alignments.

(ii) Phylogenetic trees were constructed implementing ClustalW with observed distance and Kimura correction (Thompson *et al*., 1994). Bootstrap values were used to estimate reliability of a given branching order. A total of 100 bootstrap tests were run on trees. Only bootstrap values >60% were considered to be significant.

2.3 Comparison of ortholog clusters

Ortholog clusters generated by OrthoMCL, KOG, or manually made, were compared to the output of MultiParanoid in both directions. The comparison program took each cluster (''query'') from the first set and searched for its genes in the second group of clusters. If the genes were found in more than one cluster, the query cluster was labeled as ''Split''. Query clusters with no counterparts were labeled as ''Not found''. Otherwise (exactly one cluster found), its congruity to the query was tested. A result for the query cluster was classified into a number of categories (Supplementary Table 1). For each gene clustered by only first of the compared methods, a series of possible reasons were checked (Figure 3).

3 RESULTS

3.1 The MultiParanoid algorithm

The MultiParanoid algorithm is in its default form a simple chaining together of overlapping pairwise ortholog groups. It thus depends heavily on the quality of these groups—errors here will be propagated to the multi-species clusters. We therefore used InParanoid with default parameters, which are relatively strict, to generate the pairwise groups. As MultiParanoid provides confidence scores for the cluster members, calculated as mean InParanoid scores from the pairwise clusters, we explored ways to tighten the multi-species clusters by excluding orthologs of lower confidence. However, we found that this mainly increased the false negative rate (data not shown).

It is important to keep in mind that MultiParanoid was designed to only handle multiple proteomes that all diverged at roughly the same time point. If species of unequal relatedness are clustered, e.g. yeast, human, and chimpanzee, an implicit problem is created.

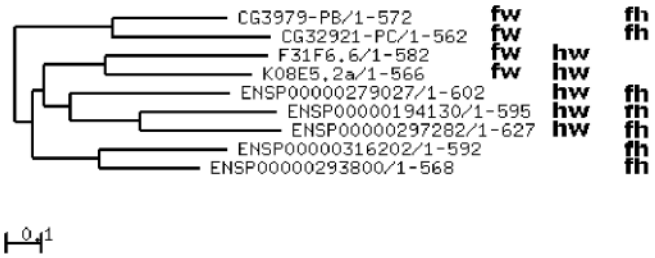


Fig. 1. Illustration of a ''tree conflict'' that may occur when merging multiple InParanoid clusters into one MultiParanoid cluster. All the sequences of the tree belong to a single MultiParanoid cluster 3575 (version 1.00), including five human proteins (ENSP*), two fly proteins (CG*), and two worm proteins. At the InParanoid 2-species level however, only ENSP00000279027, ENSP00000194130, ENSP00000297282 were recognized as human orthologs of the worm genes, yet all five were orthologous to the fly genes (InParanoid cluster members are indicated by the labels fh: fly-human; fw: fly-worm; hw: human-worm).

There is no ancestral node in the species tree of these organisms that represents the last common ancestor for all the species pairs. The resulting clusters will therefore often contain human-chimpanzee outparalogs, which were included because they are bona fide inparalogs relative to yeast.

This constitutes a major principle difference between MultiParanoid and KOGs/OrthoMCL. Both these databases combine species at very different distances, which makes the clusters less strict ortholog groups. Another difference is that only MultiParanoid gives the user confidence values. OrthoMCL is in many ways similar to InParanoid in its treatment of seed and inparalogs, but the algorithm based on Markov clustering is very different. It uses normalized E-values rather than bit scores, and the clustering is done in one step for all proteomes. A drawback is that the ''inflation parameter'' that governs the tightness of the clusters needs to be set in an ad hoc fashion.

3.2 Manual construction of the reference set

When the InParanoid algorithm was originally developed, a manually curated dataset of human-worm ortholog groups was used as a trusted standard to evaluate the accuracy of the predicted groups (Remm *et al*., 2001). Here we have extended the original dataset of human-worm orthologs by including fly orthologs to create a suitable 3-species reference set to test the accuracy of the MultiParanoid algorithm. The original dataset contained 221 groups, and most of these (202) could be extended with fly orthologs. However, in 19 cases, no fly ortholog was found, and in some cases the original group had to be redefined in the light of the fly ortholog. In total, the new reference set contains 221 groups (141 human-worm-fly, 19 human-worm, 28 human-fly, and 33 worm-fly). It is built from 697 human, 307 fly, and 361 worm proteins. This manually curated dataset is available at http://multiparanoid.cgb.ki.se/stats.html and can be used as a reference set by other developers of algorithms for detecting ortholog groups.

3.3 Benchmarking MultiParanoid

We executed MultiParanoid on the same versions of the human, fly, and worm proteomes that were used to create the reference set. To characterize MultiParanoid's ability to reconstruct the manual clusters, we extracted the intersecting and non-overlapping sets between the two clusterings, as shown in supplementary Table 1A. Both clusterings had roughly the same number of clusters: 221 in the reference set and 214 by MultiParanoid. Of these, 132 were identical. Another 45 clusters were almost identical in the sense that one was a subset of the other. This leaves about 40 clusters that clearly differed. Inspection of these cases revealed that the prevalent reason for the disagreement is the different sequence distances obtained by pairwise alignments used in InParanoid and those obtained by multiple alignments used for the manual phylogenetic analyses. Moreover, a manual curator's perception of what constitutes a ''too short'' or ''too weak'' match may differ from the strict InParanoid cutoffs.

3.4 Comparison to other methods

MultiParanoid was compared to two alternative methods: KOGs (Tatusov *et al*., 2003) and OrthoMCL (Li *et al*., 2003). To ensure a direct comparison, we ran MultiParanoid on the data used in the KOG and OrthoMCL publications. Both KOGs and OrthoMCL original clusters contained sequences of additional species, but to simplify the comparison, only sequences from human, worm, and fly were considered.

A detailed analysis was performed between MultiParanoid and the two other databases. Corresponding clusters were identified and their content was compared (see Methods). When the clusters differed, we categorized the differences into the following types: split, subset, mismatch (partial overlap), and absence. The number of clusters and genes in these categories are listed in Supplementary Table 1.

The genes that were clustered by only one of the methods were further analyzed to establish a plausible cause of discrepancy. A visual inspection of selected clusters pointed to a number of typical reasons for the observed differences. We decided to use these main categories: tree conflict, too short match, too weak match, outparalog, and other (reason not established). The classification was done in this priority order.

Tree conflict describes the case when a set of inparalogs in proteome A from the comparison A-B disagree with the inparalogs from A-C. Tree conflicts typically occur when combining species at different evolutionary distances (which thus should be avoided), or if one species has lost the original genes. A tree conflict is illustrated in Figure 1: the human-worm InParanoid clustering produced three human inparalogs while human-fly produced five. This can sometimes happen although human/fly/worm descend from roughly the same last common ancestor (of the Bilateria clade); here it was caused by a rather arbitrary clustering of the human/worm genes when the BLAST scores of the alternatives were very close. Tree conflicts are relatively common, and only result in a warning. The total number of clusters generated by MultiParanoid, run on updated human, fly, worm proteomes, that were affected by the tree conflict was 1026 of 6348 (16.1%).

Genes were classified as outparalogs when (1) a paralog (from the same species) exists in the cluster, and it is found in the corresponding cluster of the other method, and (2) a gene from another species is found closer to the second paralog than the paralogs are to each other.

The most striking difference when comparing MultiParanoid to KOGs for human/fly/worm (Supplementary Table 1A) is that although KOGs contain fewer clusters (4543 compared to 5755

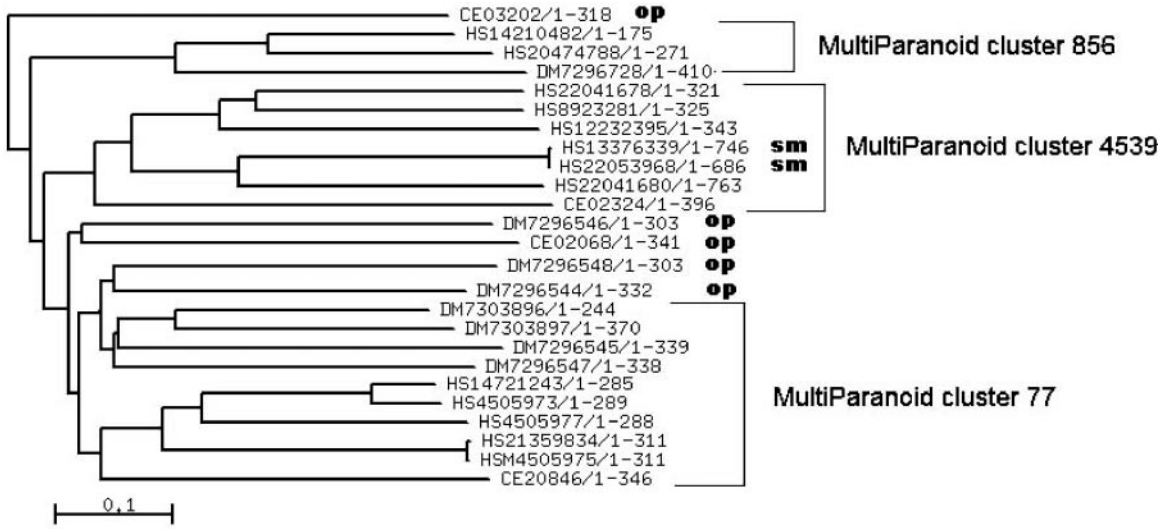


Fig. 2. Example of differences between KOG and MultiParanoid. The sequences in the tree are all the members of KOG cluster 3030. Three subtrees were identified as independent ortholog groups by MultiParanoid. HS*: human proteins, DM*: fly proteins, CE*: worm proteins. Labels: sm: sequence with a ''short match'' to the tree neighbours and therefore not clustered by MultiParanoid; op: outparalog. Note that the op-labeled fly sequences DM7296548 and DM7296544 look like inparalogs in this tree built from a multiple alignment, yet they fell just outside the cluster in InParanoid. This illustrates the clustering differences that may result from different ways of producing the sequence distance matrix.

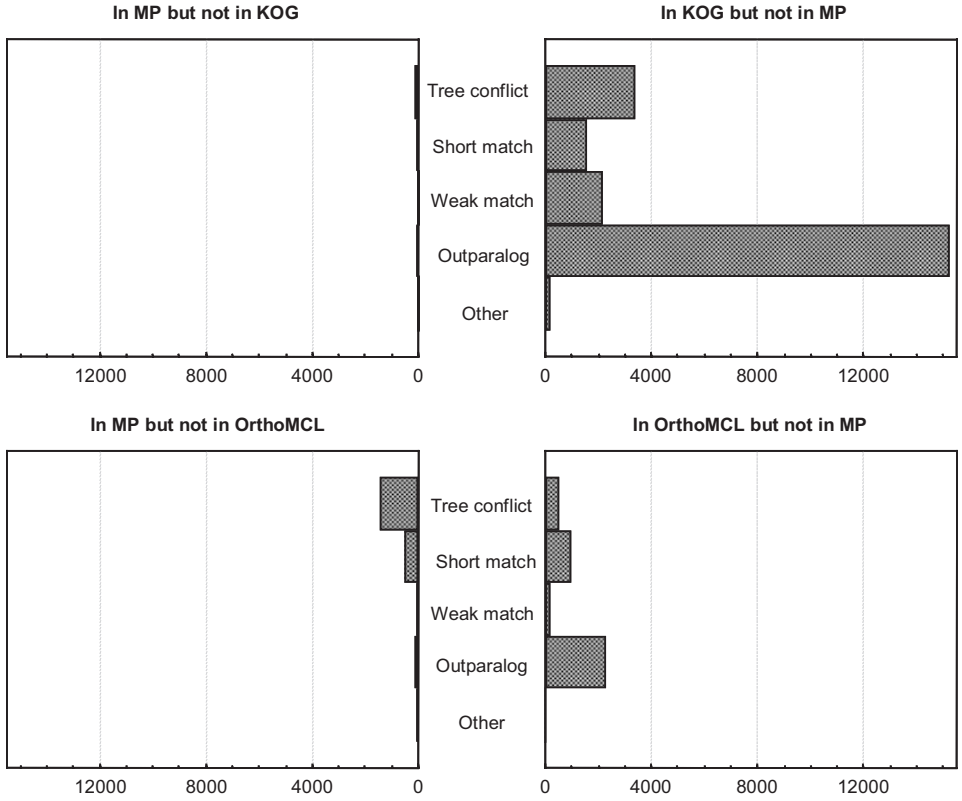


Fig. 3. Comparison of MultiParanoid to KOG and OrthoMCL. A detailed analysis was made of features and possible reasons for observed differences of corresponding ortholog clusters. Sequences clustered in one database but missing from the corresponding cluster in the other database were classified into the following categories* *Tree conflict*: conflict when merging pairwise ortholog groups in MultiParanoid (MP); *Short match:* matches the other cluster proteins with less than 50% of the length; *Weak match:* matches the other cluster proteins below the BLAST cutoff (50 bits); *Outparalog:* the protein is part of another subtree that includes the last common ancestor; *Other:* none of the reasons indicated above. *In case of multiple features per protein, only one is counted in the priority of the list above.

for MultiParanoid), they contain many more sequences (37737 compared to 23122). The average cluster size is thus twice as large in KOGs (8.3 versus 4.0). Only 1451 clusters were identical between KOG and MultiParanoid, while 2094 KOG clusters were supersets of the corresponding MultiParanoid clusters. An example of a typical situation is shown in Figure 2, in which one KOGs cluster contains three separate MultiParanoid clusters. In most of these cases it is clear that the MultiParanoid clusters represent more realistic ortholog groups in which all members derive from a single gene in the last common ancestor (of *Bilateria*). Similar cases have thus been classified as outparalogs in Figure 3. Indeed, of the 22590 KOG genes not found in the corresponding MultiParanoid cluster, 67.6% (15271) were classified outparalogs, which should be seen as an error in KOGs. The second largest reason was tree conflict (15.1%, 3411) followed by weak and short matches (9.6% and 6.9%). The latter two discrepancies may be explained by the fact that InParanoid does not accept matches below 50 bits and 50% of the length. Note that although many of the genes categorized as tree conflict probably also represent outparalogs, we chose to not classify them as such because the tree conflict casts some doubt about the whole cluster. In other words, our figures underestimate the number of outparalogs in KOGs.

The high outparalog rate in KOGs is partly due to the fact that most clusters were built with regard to a higher last common ancestor, e.g. the one of eukaryotes, and contain species beyond the animal clade. Indeed, only 1147 KOG clusters (of 4852) were animal-specific. But even when looking at 50 randomly selected pure human/fly/worm KOG clusters, 32 contained outparalogs by visual inspection of the gene trees. Many of the two-species KOG clusters (TWOGs) with >2 genes also contained outparalogs. Thus, KOGs appears to generally favor inclusion of outparalogs.

The OrthoMCL clusters were in much better agreement with the MultiParanoid results—both produced roughly 6000 clusters containing about 26000 genes. About 4000 of the clusters were identical, suggesting that these ortholog groups are very trustworthy. In the roughly 2000 clusters with differences, a couple of trends stood out. Outparalog inclusion was about 15 times more common among the OrthoMCL-unique genes (2267) compared to MultiParanoid-unique ones (145). The fact that tree conflicts are three times more common in clusters with MultiParanoid-unique genes than OrthoMCL-unique ones (1453 versus 518) suggests that OrthoMCL builds slightly tighter clusters than MultiParanoid. Genes missing due to short or weak matches were about twice as common in OrthoMCL, indicating that MultiParanoid is stricter in these respects.

The main difference between MultiParanoid and OrthoMCL thus seems to be OrthoMCL's tendency to include outparalogs. This can be explained by the fact that the original OrthoMCL clusters included 10 proteomes, some of which have very different last shared ancestors. For instance, human, mouse, and *E. coli* are included at the same time. Combining proteome pairs with such different relationships inevitably leads to inclusion of outparalogs: the eukaryotic genes underwent multiple common duplications since the divergence from *E. coli*. This problem has been worsened in the latest version of OrthoMCL, which includes 55 proteomes (Chen *et al*., 2006). For example, taking the top 10 (by E-value) OrthoMCL clusters that contained >8 genes from human, *Ciona*, *D. melanogaster*, and *C. elegans*, 8 clusters (111, 1057, 489, 88, 1300, 335, 1428, 123) contained outparalogs in at least 1 species (usually in 2-4). In the previous 10-species OrthoMCL version, only cluster 1057 (rather its prototype, as the numbering was changed) had outparalogs. The corresponding MultiParanoid 4-species clusters had no outparalogs.

Another database of eukaryotic orthologs is HomoloGene (Wheeler *et al*. 2006), which in addition to sequence similarity also uses synteny and DNA substitution rates to build ortholog groups (http://www.ncbi.nlm.nih.gov/HomoloGene/HTML/homologene_buildproc.html). This database is however very different in nature from MultiParanoid, OrthoMCL and KOGs. HomoloGene is extreme in the opposite way that KOGs is—it splits up ortholog groups into smaller groups, putting inparalogs into different clusters.

For example, only 29 of 3814 HomoloGene clusters that could include both human and yeast genes (labeled ''*Eukaryota*'' or ''*Fungi*/*Metazoa*'') contained more than a single human gene. As a comparison, InParanoid had 2138 human-yeast clusters, and 816 of them contained more than one human orthologs.

Genes that are considered inparalogs by InParanoid are normally not missing from HomoloGene but are found in other clusters, usually with different labeling of the last common ancestor (e.g. human inparalogs could be in clusters labeled ''*Eukaryota*'', ''*Coelomata*'', ''*Amniota*'') or with the same label but another species content.

For instance, the biggest MultiParanoid cluster built from human, *Ciona*, *D. melanogaster*, and *C. elegans* proteins contained more than 400 human genes (zinc finger proteins with Pfam domain zf-C2H2, PF00096), but only a few from other species. The human part thus constituted a vertebrate-specific expansion according to MultiParanoid. Yet, in HomoloGene most of the human genes were split into 6 different clusters labeled higher than vertebrates (''*Coelomata*'' and ''*Fungi*/*Metazoa*''). These clusters contained a set of human genes plus an insect or worm gene (all from the same MultiParanoid cluster), even though the human genes are closer to each other than to any gene outside the vertebrate clade. Some human genes from the MultiParanoid cluster were placed in pure vertebrate clusters (''*Amniota*'', ''*Eutheria*'', ''*Euarchontoglires*'', and human-specific expansions).

HomoloGene thus tends spread inparalogs over isolated small clusters. This property makes the clusters very tight, practically inparalog-free, and misleading in defining complete ortholog sets.

4 DISCUSSION

Functional genomics has driven a demand for fast and efficient orthology analysis tools. The algorithm presented here enables an automated orthology analysis to be performed on multiple proteomes, and is therefore a welcome extension of the previously published InParanoid (Remm *et al*., 2001) algorithm. We found a satisfying high degree of congruence between the results generated by MultiParanoid and the manually curated dataset used as a reference. The ability of the algorithm to correctly identify orthologous sequences was also evaluated by executing MultiParanoid and similar algorithms published by other groups, namely KOGs (Tatusov *et al*., 2003) and OrthoMCL (Li *et al*., 2003), on the same datasets. This showed that the quality of MultiParanoid's clusters is high, and therefore the method should make an important contribution to the bioinformatics tools currently available for orthology analyses.

Unlike KOGs where the minimal cluster consists of three genes, one per species (''triangles'', Tatusov *et al*., 1997), MultiParanoid is based on pairwise groups of orthologs. For genomes A, B, C, protein pairs {A1, B1} and {B1, C1} can be reciprocally best hits, whereas {A1, C1} may not be. Hence, clusters exist where a triangle is not secured. This leads to what we call a ''tree conflict'' when merging pairwise orthologs from three species. If the species have roughly the same last ancestor we believe that the best action in such cases is to combine all genes from the pairwise clusters. Still, only a minor fraction (∼15%) of MultiParanoid clusters had tree conflicts when clustering human, fly, and worm. In very few cases (100-200 genes) did the conflict lead to ambiguous cluster membership.

We here consider human, fly, and worm to descend from roughly the same last ancestor. Yet, two different subgroupings have been proposed: the ''*Ecdysozoa*'' (worm-fly) and ''*Coelomata*'' (fly-human) hypotheses (Blair *et al*., 2002; Dopazo and Dopazo 2005; Philip *et al*., 2005). Neither of these gets full support from molecular data. Looking at gene trees, the *Coelomata* grouping is found in about 60% of the trees, *Ecdysozoa* in 25%, and worm-human in 15%. The question is therefore probably unresolvable and we consider the three species to be roughly equally related. It is thus wiser to use molecular data to group species than to use the classical taxonomy, especially since the latter can be ambiguous or vague.

The requirement of only clustering species with shared last ancestor can be a drawback for MultiParanoid, as it only allows few eukaryotic species to be included in multi-species groups. However, a possibility is to consider several species in a clade as a 'pseudo-species', e.g. mammals or arthropods. If one treats all mammalian genes as 'pseudo-inparalogs' when compared to arthropods and nematodes, it is possible to avoid outparalogs. This is done by labeling the included outparalogs as pseudo-inparalogs, and not transferring functional information between them. We are developing a new version of MultiParanoid with multiple species in the same clade with precise labeling of what are orthologs and what are not within each cluster. Using this framework, which is similar to the HOPS database (Storm and Sonnhammer, 2003), we can build clusters that include all completely sequenced eukaryotic species. Even incomplete proteomes can be included, as long as one complete proteome is part of the clade.

5 DATA

The manually curated data set of transmembrane proteins was based on the older proteome versions:

Human: 35118 sequences from SwissProt and TREMBL.

Fly: 14100 predicted proteins sequences from FlyPep Release 1. (http://www.fruitfly.org/sequence/download.html).

Worm: 19099 predicted proteins from WormPep 20 (ftp://ftp.wormbase.org/pub/wormbase/).

These original protein sets and the manually curated clusters are available at http://multiparanoid.cgb.ki.se/download.

The KOG clusters were published as a supplementary material by Tatusov *et al*. (2003) and were downloaded at http://www.ncbi.nlm.nih.gov/COG/new/. For the purpose of this work, only human, fly and worm genes were extracted from the seven species in total. In addition, we included the 2-species clusters (TWOGS). The version numbers of the proteomes are not available, but the original complete sets of proteins of all KOG proteomes in FASTA format can be downloaded from the same location.

OrthoMCL clusters were gathered via queries to the Web service http://www.cbil.upenn.edu/gene-family/. The following datasets were downloaded: {human, fly, worm}, {human, fly}, {human, worm}, {fly, worm}. The respective versions of complete protein sets were obtained from the original sites listed in their article (Li *et al*., 2003).

Alternative splice forms of the same gene may sometimes end up in different clusters. We therefore only used the longest spliced form of each gene.

MultiParanoid scripts, FASTA sequence and data files are available from the web site http://multiparanoid.cgb.ki.se/. The final clusters generated by MultiParanoid can be downloaded as a single text file. The web-based version 1.00 of MultiParanoid with search by gene/protein ID and cross-links to protein/domain databases currently includes four genomes—*C. elegans*, *D. melanogaster*, *C. intestinalis*, and *H. sapiens*—and will be expanded.

ACKNOWLEDGEMENTS

This work was supported by a grant from Pfizer Corporation.

REFERENCES

Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **25,** 3389-3402.

Blair,J.E., Ikeo,K., Gojobori,T. and Hedges,S.B. (2002) The evolutionary position of nematodes. *BMC Evol. Biol.* **2,** 7.

Bono,H., Goto,S., Fujibuchi,W., Ogata,H. and Kanehisa,M. (1998) Systematic prediction of orthologous units of genes in the complete genomes. *Genome Inform Ser Workshop Genome Inform.* **9**, 32–40.

Chen,F., Mackey,A.J., Stoeckert,C.J.Jr and Roos,D.S. (2006) OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Res.* Jan 1; **34,** (Database issue):D363–8.

Chervitz,S.A., Aravind,L., Sherlock,G., Ball,C.A., Koonin,E.V., Dwight,S.S., Harris,M.A., Dolinski,K., Mohr,S., Smith,T., Weng,S., Cherry,J.M. and Botstein,D. (1998) Comparison of the complete protein sets of worm and yeast: orthology and divergence. *Science*, Dec 11;**282**(5396), 2022–8.

Dopazo,H and Dopazo,J. (2005) Genome-scale evidence of the nematode-arthropod clade. *Genome Biol.*; **6**(5), R41. Epub 2005 Apr 28.

Enmark,E., Gustafsson,J.A. (2001) Comparing nuclear receptors in worms, flies and humans. *Trends Pharmacol Sci*. **22**(12), 611–5. Review.

Fitch,W.M. (1970) Distinguishing homologous from analogous proteins. *Syst. Zool*. **19**, 99–113.

Fitch,W.M. (2000) Homology: A personal view on some of the problems. *Trends Genet.* **16**, 227–231.

Kondrashov,F.A., Rogozin,I.B., Wolf,Y.I., Koonin,E.V. (2002) Selection in the evolution of gene duplications. *Genome Biol.*, **3**(2), RESEARCH0008. Epub 2002 Jan 14.

Lee,Y., Sultana,R., Pertea,G., Cho,J., Karamycheva,S., Tsai,J., Parvizi,B., Cheung,F., Antonescu,V., White,J., Holt,I., Liang,F. and Quackenbush,J. (2002) Cross-referencing eukaryotic genomes: TIGR Orthologous Gene Alignments (TOGA). *Genome Res.* **12**(3), 493–502.

Lespinet,O., Wolf,Y.I., Koonin,E.V. and Aravind,L. (2002) The role of lineage-specific gene family expansion in the evolution of eukaryotes. *Genome Res.* **12**(7), 1048-59.

Li,L., Stoeckert,C.J., Roos,D.S. (2003) OrthoMCL: Identification of ortholog groups for eukaryotic genomes *Genome Res.* **13**(9), 2178–89.

Maglich,J.M., Sluder,A., Guan,X., Shi,Y., McKee,D.D., Carrick,K., Kamdar,K., Willson,T.M. and Moore,I.T. (2001) Comparison of complete nuclear receptor sets from the human, *Caenorhabditis elegans* and *Drosophila* genomes. *Genome Biol*. **2**(8), RESEARCH0029. Epub 2001 Jul 24.

Mushegian,A.R., Garey,J.R., Martin,J. and Liu,L.X. (1998) Large-scale taxonomic profiling of eukaryotic model organisms: A comparison of orthologous proteins encoded by the human, fly, nematode, and yeast genomes. *Genome Res.* **8**, 590-598.

Philip,G.K., Creevey,C.J. and McInerney,J.O. (2005) The *Opisthokonta* and the *Ecdysozoa* may not be clades: stronger support for the grouping of plant and animal than for animal and fungi and stronger support for the *Coelomata* than *Ecdysozoa*. *Mol. Biol. Evol.* **22**(5), 1175-84. Epub 2005 Feb 9.

Remm,M., Storm,C.E. and Sonnhammer,E.L.L. (2001) Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.* **314**, 1041–1052.

Rubin,G.M., Yandell,M.D., Wortman,J.R., Gabor Miklos,G.L., Nelson,C.R., Hariharan,I.K., Fortini,M.E., Li,P.W., Apweiler,R., Fleischmann,W., Cherry,J.M., Henikoff,S., Skupski,M.P., Misra,S., Ashburner,M., Birney,E., Boguski,M.S., Brody,T., Brokstein,P., Celniker,S.E., Chervitz,S.A., Coates,D., Cravchik,A., Gabrielian,A., Galle,R.F., Gelbart,W.M., George,R.A., Goldstein,L.S., Gong,F., Guan,P., Harris,N.L., Hay,B.A., Hoskins,R.A., Li,J., Li,Z., Hynes,R.O., Jones,S.J., Kuehl,P.M., Lemaitre,B., Littleton,J.T., Morrison,D.K., Mungall,C., O'Farrell,P.H., Pickeral,O.K., Shue,C., Vosshall,L.B., Zhang,J., Zhao,Q. and Zheng,X.H. (2000) Comparative genomics of the eukaryotes. *Science*, Mar. 24 **287**(5461), 2204–15.

Sonnhammer,E.L.L. and Koonin,E.V. (2002) Orthology, paralogy and proposed classification for paralog subtypes. *Trends Genet.* **18**, 619-620.

Storm,C.E. and Sonnhammer,E.L.L. (2003) Comprehensive analysis of orthologous protein domains using the HOPS database. *Genome Res.* **13**(10), 2353-62.

Tatusov,R.L., Koonin,E.V. and Lipman,D.J. (1997) A genomic perspective on protein families. *Science*, **278**, 631–637.

Tatusov,R.L., Fedorova,N.D., Jackson,J.D., Jacobs,A.R., Kiryutin,B., Koonin,E.V., Krylov,D.M., Mazumder,R., Mekhedov,S.L., Nikolskaya,A.N., Rao,B.S., Smirnov,S., Sverdlov,A.V., Vasudevan,S., Wolf,Y.I., Yin,J.J. and Natale,D.A. (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*. Sep 11; **4**(1), 41.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* Nov 11; **22**(22), 4673–80.

Troemel,E.R., Chou,J.H., Dwyer,N.D., Colbert,H.A. and Bargmann,C.I. (1995) Divergent seven transmembrane receptors are candidate chemosensory receptors in *C. elegans*. *Cell*, Oct 20; **83**(2), 207–18.

Wheelan,S.J., Boguski,M.S., Duret,L., Makalowski,W. (1999) Human and nematode orthologs – lessons from the analysis of 1800 human genes and the proteome of *Caenorhabditis elegans*. *Gene*, Sep 30; **238**(1), 163–70.

Wheeler,D.L., Barrett,T., Benson,D.A., Bryant,S.H., Canese,K., Chetvernin,V., Church,D.M., DiCuccio,M., Edgar,R., Federhen,S., Geer,L.Y., Helmberg,W., Kapustin,Y., Kenton,D.L., Khovayko,O., Lipman,D.J., Madden,T.L., Maglott,D.R., Ostell,J., Pruitt,K.D., Schuler,G.D., Schriml,L.M., Sequeira,E., Sherry,S.T., Sirotkin,K., Souvorov,A., Starchenko,G., Suzek,T.O., Tatusov,R., Tatusova,T.A., Wagner,L. and Yaschenko,E. (2006) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, Jan 1; **34** (Database issue):D173–80.

Xie T. and Ding D. (2000) Investigating 42 candidate orthologous protein groups by molecular evolutionary analysis on genome scale. *Gene,* **261**, 305–310.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e16–e24
doi:10.1093/bioinformatics/btl216

DynaPred: A structure and sequence based method for the prediction of MHC class I binding peptide sequences and conformations

Iris Antes[1,*], Shirley W. I. Siu[1] and Thomas Lengauer

MPI für Informatik, Stuhlsatzenhausweg 85, D-66123 Saarbrücken, Germany

ABSTRACT

Motivation: The binding of endogenous antigenic peptides to MHC class I molecules is an important step during the immunologic response of a host against a pathogen. Thus, various sequence- and structure-based prediction methods have been proposed for this purpose. The sequence-based methods are computationally efficient, but are hampered by the need of sufficient experimental data and do not provide a structural interpretation of their results. The structural methods are data-independent, but are quite time-consuming and thus not suited for screening of whole genomes. Here, we present a new method, which performs sequence-based prediction by incorporating information obtained from molecular modeling. This allows us to perform large databases screening and to provide structural information of the results.

Results: We developed a SVM-trained, quantitative matrix-based method for the prediction of MHC class I binding peptides, in which the features of the scoring matrix are energy terms retrieved from molecular dynamics simulations. At the same time we used the equilibrated structures obtained from the same simulations in a simple and efficient docking procedure. Our method consists of two steps: First, we predict potential binders from sequence data alone and second, we construct protein-peptide complexes for the predicted binders. So far, we tested our approach on the HLA-A0201 allele. We constructed two prediction models, using local, position-dependent (*DynaPred*^{POS}) and global, position-independent (*DynaPred*) features. The former model outperformed the two sequence-based methods used in our evaluation; the latter shows a much higher generalizability towards other alleles than the position-dependent models. The constructed peptide structures can be refined within seconds to structures with an average backbone RMSD of 1.53 Å from the corresponding experimental structures.

Contact: antes@mpi-sb.mpg.de

1 INTRODUCTION

The binding of antigenic peptides originating from pathogens to the major histocompatibility complex (MHC) class I is one of the crucial steps during the intracellular immunological response against the intruder (Paul *et al*., 1998). After a pathogen enters the host cell, proteins from the invading organism are cleaved into smaller peptide fragments by the proteasome. These fragments are transported into the endoplasmic reticulum by the TAP proteins, where they bind to MHC molecules. Afterwards the MHC-peptide complex is translocated to the cell surface. At the surface of the cell, pathogenic peptides are identified by T-cell receptors (TCRs) via TCR-MHC-peptide complex formation. This step initiates the immunological response against the pathogen. Peptides which can trigger such a response are called epitopes. Not all peptides binding to MHC molecules are epitopes, but all T-cell epitopes need to bind to MHC molecules. Thus, knowing which and understanding why certain peptides bind to a specific MHC is not only fundamental to the understanding of the immune system, but also a crucial step in vaccine and immunotherapeutic development. Experimental screening of peptides with respect to their MHC binding capabilities is very demanding due to the large number of possible peptide sequences and the high polymorphism of the MHC molecules. Thus there is a strong interest in computational methods for predicting the binding capabilities of peptides to MHC as a first step to select peptides for screening.

For the prediction of MHC (class I and II) binding peptides, sequence- and structure-based methods as well as their combinations were used for both classification and regression models. Classification models distinguish binders from non-binders, whereas regression methods try to predict the binding affinity of peptides to MHC molecules.

Sequence based prediction methods include binding motifs (Rammensee *et al*., 1999; Hammer, 1995; Reche *et al*., 2002; Peters *et al*., 2003), quantitative matrices (Parker *et al*., 1994; Southwood *et al*., 1998), data-derived matrices (Yu *et al*., 2002), and the combination of a motif based approach with Gibbs sampling (Nielsen *et al*., 2004). For the training, various machine learning techniques have been applied such as artificial neural networks (Brusic *et al*., 1998; Gulukota *et al*., 1997; Milik *et al*., 1998), hidden markov models (Mamitsuka 1998), classification trees (Segal *et al*., 2001), support vector machines (Dönnes and Elofsson, 2002; Zao *et al*., 2003; Bhasin *et al*., 2004), and biosupport vector machines (Yang and Johnson, 2005). These methods encode sequences as binary vectors or as numerical vectors based on their physiochemical property values. Due to the limited public availability of consistent quantitative binding data, most methods are trained for classification. Still, regression was performed so far in QSAR studies

*To whom correspondence should be addressed.
[1]Both authors contributed equally to this work.

(Doytchinova *et al*., 2002, 2004; Li *et al*., 2004) and using average relative binding matrices (Bui *et al*., 2005). Structural information has been used for prediction in the context of 3D-QSAR (Doytchinova *et al*., 2002, 2004) and docking (Bordner and Abagyan, 2006).

Most prediction methods are based on the so called 'additive model'. This model assumes that the overall binding affinity of a peptide can be approximated as the sum of the properties of the individual residues. Extensions of this model by including neighbor interactions have led only to slight or even no improvement of the prediction accuracy (Doytchinova *et al*., 2002; Peters *et al*., 2003). In the context of 3D-QSAR the additive model was compared to a model based only on 'global' structural features (Doytchinova *et al*., 2005), which were calculated for the whole peptide and not for the individual residues. This study showed that global features did not perform as well as local, residue based features for binding affinity prediction. The success of the additive model can be explained by the structure of the MHC binding groove, which consists of nine residue binding pockets located next to each other along the groove. The peptide is bound in an extended conformation with one residue of the peptide occupying exactly one binding pocket, thus the effect of the interaction between the neighboring side chains is minimal.

Several structural search algorithms for the identification of low energy peptide-binding conformations have been proposed. One class of methods is based on the observation that for each MHC allele there are certain conserved peptide 'anchor' residues which bind tightly to specific MHC binding pockets. These approaches (Rosenfeld *et al*., 2003, Tong *et al*., 2004, Logean *et al*., 2002) consist of two main steps: first, placing the anchor residues in the binding pocket and second, constructing the rest of the peptide based on the anchor positions. A different class of methods is based on the division of the peptides into backbone and side chains (Ota *et al*., 2001, Altuvia *et al*., 1995, Schueler-Furman *et al*., 1998). These methods use backbone conformations from experimental structures and predict the side chain conformations either by threading or the use of rotamer libraries. Another study uses dead-end elimination within a combinatorial build-up algorithm (Desmet *et al*., 1997). The method, which is closest to our proposed method, is a residue-based free-energy mapping approach (Sezermann *et al*., 1996). Two other studies use Monte-Carlo annealing approaches to dock peptides into the binding pocket (Liu *et al*., 2004) and use the docking scores for prediction (Bordner *et al*., 2006).

Comparing sequence and structure-based methods, the latter have the advantage that they are independent of the amount of experimental binding data, but are too time-consuming for the screening of large numbers of peptides. On the other hand, sequence-based prediction methods are fast, but are strongly dependent on the amount of binding data available for specific alleles. Thus currently they achieve high performance only for the intensively investigated alleles. This becomes even more serious for the quantitative prediction of binding affinities because for this purpose large screening experiments are necessary to produce comparable IC_{50} values for the training of the models. Although such efforts are ongoing, they will always be focused towards the most important alleles. Another drawback of sequence based methods is their limited structural interpretability, which is of crucial importance for the design of peptide mimicking vaccines and drug like molecules.

Here we present a combined two-step structure and sequence-based prediction method *DynaPred*, which allows at the same time a fast prediction of MHC class I binders and an efficient construction of docked peptide conformations. The prediction method uses two feature matrices derived from structural calculations as basis for support vector machine training: A local, position-dependent ($DynaPred^{POS}$) and a global, position-independent (*DynaPred)* matrix. The docking method is based on equilibrated, pre-calculated structures for each amino acid in each of the binding pockets. So far quantitative matrices used for the prediction of MHC-binding peptides are based on sequence data, partially including biophysical amino acid properties. Structure-based biophysical data were used in the context of 3D-QSAR, which, however, only considers the structural properties of the peptides, but not their interactions with the binding pocket. We based the choice of our scoring-matrix features on the linear energy approximation for the calculation of binding affinities. Linear energy models (Aqvist *et al*., 2002) were used in various studies and were successfully tested for predicting the binding affinities of tri-peptides to OppA (Wang *et al*., 2002). In the context of MHC-peptide binding, such approaches were applied for the scoring of docked peptides (Logean *et al*., 2002, Sezermann *et al*., 1996) and prediction based on docking results (Bordner *et al*., 2006).

However, to our knowledge this is the first time that structure-based interaction energy terms are used for a residue-based prediction approach for peptide binding. A residue-based docking method was presented for MHC-peptide complexes by Sezermann *et al*., 1996. However, its discrete rotamer-based search algorithm leads to many different peptide structures, all very similar in energy, and thus extensive post-processing of these structures is necessary to find the best conformer. We avoid this last step by the use of one equilibrated residue side chain conformation, which was calculated by molecular dynamics, instead of a discrete search algorithm.

We implemented and evaluated our approach for the most frequently occurring allele HLA-A*0201 with 9-mer peptides.

2 METHODS

2.1 General strategy

The basic strategy behind our method is to approximate the binding free energy of all 20 amino acids in each of the nine binding pockets of the MHC binding groove using energetic information obtained by molecular dynamics simulations. This information is used subsequently for the training of a sequence-based predictive model. In addition, the structural information obtained by the simulations is used for constructing the peptide-protein complexes of the predicted binders. Our algorithm is based on a single main assumption: The total binding affinity of a peptide can be approximated as the sum of the binding affinities of its individual amino acids, neglecting the effect of the neighboring residues (See 'Introduction' for the validity of this assumption). This allows us to simulate each amino acid individually in each binding pocket. Initial conformations of the individual residues bound to the MHC protein are constructed from crystal structures. To stabilize the peptide conformations, we extend the single residues to peptide-trimers and dimers, by adding a glycine residue at both sides (for terminating residues only on the non-terminating side). For side chains for which no bound conformation was available, existing residues are mutated to the corresponding amino acid. MD simulations are performed on the bound complexes as well as on the individual molecules in solution. Important energy terms reflecting the binding properties of the amino acids are calculated from the simulation results and subsequently used for the construction of a binding-free-energy-based scoring matrix (BFESM). This matrix contains energy terms for each residue in each binding pocket and forms the basis for the construction of two prediction models.

Our approach allows us to predict MHC-binding peptides with the speed of sequence-based methods, but on the basis of structurally derived energies. In addition, the equilibrated MD structures serve as templates to enable fast construction of the conformations predicted binding sequences.

Our proposed method can be summarized as follows:

(1) For a given MHC allele, the compatibility of each amino acid in each of the nine binding pockets is examined thoroughly by MD simulations.

(2) A Binding-Free-Energy-Based Scoring Matrix (BFESM) is produced by extracting values of energy terms important for binding from the simulations.

(3) The position-based bound conformations are extracted from the simulations for each amino acid type and saved in a data base.

(4) Experimental binding data together with the BFESM is used in the training process to generate the prediction models.

Finally, prediction is a two step process: First, the query sequence is classified as a binder or non-binder; then the bound conformation of a predicted binder is generated.

2.2 Scoring matrix

For the construction of the Binding-Free-Energy-based Scoring Matrix (BFESM) we use energy terms obtained by molecular dynamics simulations. According to the linear energy model (Aqvist *et al*., 2002), the binding free energy can be approximated by the difference between the interaction energies ΔG^{el} and ΔG^{np} of the ligand in the protein-ligand complex (bound state) and in solution (free state). We extend this model by adding the energy contributions of the protein and ΔG^{int} and $T\Delta S^{conf}$:

$$\Delta G^{bind} = \Delta G^{el} + \Delta G^{np} + \Delta G^{int} - T\Delta S^{conf} \quad (1)$$

(ΔG^{el} = electrostatic, ΔG^{np} = nonpolar, ΔG^{int} = internal, $T\Delta S^{conf}$= entropic contribution)

Thus the following energy terms are included in the BFESM:

(1) The electrostatic contribution, which consists of the electrostatic interaction energy between the peptide and the MHC molecule and a desolvation term:

$$\Delta G^{el} = \langle V^{el}_{bound.p-l}\rangle + (\langle V^{el}_{bound.p-sol}\rangle - \langle V^{el}_{free.p-sol}\rangle) + (\langle V^{el}_{bound.l-sol}\rangle - \langle V^{el}_{free.l-sol}\rangle) \quad (2)$$

(p = protein, l = ligand, sol = solvent, V^{el} = electrostatic energy)

(2) The non-polar (hydrophobic) contribution, which can be approximated by change in the Solvent Accessible Surface area (SAS) upon binding:

$$\Delta G^{np} \propto \Delta SAS \quad (3)$$

A change in the surface area by 1 Å^2 corresponds to approximately 10.45 kJ mol^{-1} (Chothia, 1974). The change in SAS can be calculated as the difference in surface area between the complex and its individual components in solution.

(3) Due to the restricted space in the binding pocket, the residue might be forced to adopt a higher-energy conformation in the binding pocket than in the solvent. This effect is accounted for by the differences in the bond angle and torsion energies between the free and the bound states:

$$\Delta G^{int} = (\langle V^{int}_{bound.p}\rangle - \langle V^{int}_{free.p}\rangle) + (\langle V^{int}_{bound.l}\rangle - \langle V^{int}_{free.l}\rangle) \quad (4)$$

(4) The loss in conformational entropy, $-T\Delta S^{conf}$, can be approximated using the empirical scale of Pickett and Sternberg (Pickett and Sternberg, 1993). This model assumes that a solvent-exposed side chain, whose relative accessibility (RA) is greater than 60%, can rotate freely; whereas a buried side chain (RA < 60%) is restrained to one rotamer. The RA is defined as:

$$RA = \frac{SAS^{sc}_{bound.l}}{SAS^{sc}_{free.l}} \quad (5)$$

The correspondence between RA and its energetic contribution was taken from (Pickett and Sternberg, 1993).

In summary, to estimate the change in free energy, the energy values at the right hand side of Eq. (2)—Eq. (5) are required. They are calculated for each amino acid in each binding pocket from the MD simulations and used to construct the BFESM.

Table 1. Crystal structures used for the initial backbone conformations of the pseudo-peptides.

PDB	Peptide Source	Sequence	Res. (Å)
1AKJ	HIV reverse transcriptase	ILKEPVHGV	2.65
1DUZ	HTLV-1 TAX protein	LLFGYPVYV	1.80
1HHG	HIV-1 GP120 envelope protein	TLTSCNTSV	2.60
1QRN	Altered HTLV-1 TAX peptide P6A	LLFGYAVYV	2.80

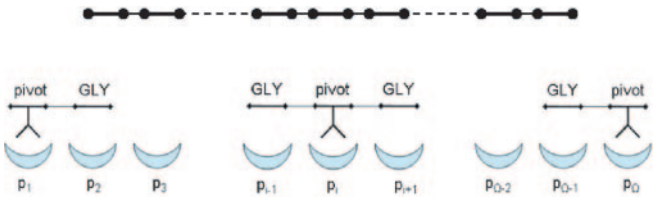


Fig. 1. Schematic representation of the pseudo-peptides used in the simulations. 3-mer or 2-mer pseudo-peptides are constructed depending on the pockets position (from p_1 to p_Ω).

2.3 Simulation setup

To calculate all energy contributions, simulations of all pseudo-peptide MHC complexes and of the MHC molecule and all amino acids in solution were performed. For the construction of the pseudo-peptides the PDB structures given in Table 1 were used. The structure of the MHC protein was taken from PDB structure 1AKJ. Each energy value is calculated as the ensemble average over the last 200ps of the trajectory after the system equilibrium is reached.

2.3.1. Pseudo-peptide generation The amino acid to be investigated (called the *pivot* residue) is embedded inside a short peptide (called *pseudo-peptide*), which is either a 2-mer or 3-mer (see Fig. 1). 2-mers are used for residues at the N and C-termini of the peptide, binding to the pocket 1 and 9. In 2-mers the pivot residue has one neighboring glycine residue. For all other binding pockets, 3-mers are used, consisting of the pivot residue and two neighboring glycines.

For the pseudo-peptide construction all structures in Table 1 were superimposed with respect to the MHC backbone surrounding the binding pocket (residue 1-180). The initial backbone conformations of the pseudo-peptides were extracted from these structures. For this purpose the bound peptide conformations were divided into di/trimers and the side chains of the first and last residue of the di/trimer were replaced by hydrogen, resulting in the two flanking GLY residues. For amino acids for which no experimental structures were available, we mutated existing residues using the program SCWRL3.0 (Canutescu *et al*., 2003).

2.3.2 Simulation conditions All molecular dynamics simulations were performed using GROMACS3.2 (Lindhal *et al*., 2001) and the OPLSAA/L force field and explicit SPC water. Long range electrostatic interactions were calculated using the Particle-Mesh Ewald method and bond constraints were applied using LINCS, and the time step was set to 2 fs. For each simulation, first a steepest-descent energy minimization was performed for 1000 steps. Then the system was solvated using a cubic box with a minimum distance of 0.7 nm between the box boundaries and the protein. The system was heated up from 0 to 300K in 100ps, before it was equilibrated at 300K using NPT ensemble (Berendsen thermo- and barostat). The total equilibration times were dependent on the flexibility of the side chains (800–3000ps). After equilibrium was reached the simulations were continued for another 200–400ps.

To approximate the constraining force of the remaining fragments of the 9-mer peptide on the pivot residue, we applied position restraints to certain atoms of the peptide during the simulations. The restraints were chosen such that the pseudo-peptide backbone was still able to move within a few Å to span the space occupied by the different backbones of the structures in Table 4 and to allow free rotation of the pivot residue. Thus, strong forces (1000 kJ/(mol*nm)) were applied only to the heavy atoms of the flanking glycine residues and weak forces (100 kJ/(mol*nm)) to the C- and N-backbone atoms of the pivot residue.

2.4 Training and testing of the prediction models

2.4.1 Binding-Free-Energy-based Scoring Matrix The Binding-Free-Energy-based Scoring Matrix (BFESM) is a quantitative matrix of dimension 20 *x (no-of-pockets) x (no-of-features*). Each entry represents one feature of a particular amino acid in a particular binding pocket. The BFESM is used to generate the feature vectors for each given sequence in the training set, all vectors together produce the feature matrix for model generation and prediction.

2.4.2 Prediction models Two feature matrices were constructed from the BFESM: A local feature matrix, which uses all the residue and binding pocket positional information from the scoring matrix and is thus called 'the position-dependent feature set' (*DynaPred*^{POS}), and a global feature matrix, for which the information from the BFESM is reduced, assuming that the positional information can be neglected and that the same feature can be summed up over all residues to give one value for each feature for each peptide. This model, the 'position-independent feature set' (*DynaPred*), can best be compared to the global features used in (Doytchinova *et al*., 2005; Bordner and Abagyan, 2006). Both features sets were tested. For the training of the support vector machines a radial kernel function were applied. The models were implemented in R (R Devel. Core Team, 2005).

2.4.3 Data sets Two publicly available data sets were used in our study: MHCPEP and SYFPEITHI. MHCPEP is a static database of MHC peptide sequences (Brusic *et al*., 1998a). Non-binding data was obtained from the author upon request. SYFPEITHI (Rammensee *et al*., 1999) is an online database with over 4500 sequences and 250 motifs from naturally processed peptides and T-cell epitopes. Since we have focused on binary classification, all 9-mer binding sequences are considered as binders, regardless of their binding specificity. Duplicated or contradicting entries were removed. Since SYFPEITHI contains only binders, the non-binding sequences from the MHCPEP data sets were included for prediction. The training of the two models was performed on three data set combinations: MHCPEP (binder + non-binder), SYFPEITHI (binder) + MHCPEP (non-binder), and MHCPEP (binder + non-binder) + SYFPEITHI (binder).

2.4.4 Testing and Evaluation We evaluated the overall performance of the prediction models, the robustness against data set size, and the generalizability with respect to other alleles. To evaluate the overall performance leave-one-out cross validation was used. To test the robustness a certain number of sequences was drawn randomly from the MHCPEP data set and each model was tested on this data set by performing 10-fold cross validation. In order to obtain an average accuracy that reflects the performance of the method in that setting, we repeated the 10-fold cross validation 10 times.

Table 2. Data sets (HLA-A*0201 allele) used in this study.

Data set	Binders	Non-binders
MHCPEP	344	383
SYFPEITHI	243	0

To test the generalizability of the prediction models, binding sequences were extracted for other HLA-A-type alleles than A*0201 from the MHCPEP database. To ensure that the results are statistically reasonable, we selected only alleles for which more than 10 unseen sequences (sequences not in the HLA-A*0201 training set) were found in the data base. We collected data for 12 alleles; 6 of them were not subtype-specific.

For evaluation on an independent data set, we chose the HIV-genome and used the prediction models trained on the combined MHCPEP/SYFPEITHI data set. For the prediction we used the complete HIV genome of the HXB2 strain (GenBank accession number K03455). The 3150 residues were divided into MHC binding and non-binding regions according to the HIV-Epitope map (Korber *et al*., 2005). Binding sequences were extracted as indicated on the map, while the non-binding sequences were generated by chopping 9-mer sequences (with eight overlapping positions) from the non-binding regions, and deleting the duplicated entries. Only peptides for which all 9 residues were located in either region (epitope or non-epitope) were included in the performance evaluation.

2.5 Construction of peptide conformations

For the construction of the peptide conformations we calculated the average conformations of the pivot residues from the last 200ps of the simulations. To generate the docked conformation of the peptide, the saved conformations of each residue in the peptide sequence were linked together inside the MHC-binding pocket of PDB structure 1AKJ (same structure as used for the simulations). Then steepest-descent energy minimization was applied to relax first the backbone and then side chains of the peptide. Afterwards the potential energies of the energy-minimized peptide structures were compared to the potential energies of the corresponding experimental structures and the RMSD of the two was calculated.

3 RESULTS

3.1 Simulation results

Simulations were performed for all pseudo-peptide-MHC complexes and the free molecules in solution. One major concern about the use of molecular dynamics for the purpose of sampling side chain conformations is that, due to the limited ability of the MD approach to cross conformational barriers, the conformational space of the residue might not be sampled adequately. We observed that for cases in which the binding pocket size allowed changes in the conformations of the side chains, these conformational changes occurred during the equilibration period of the simulations (leading to all-atom side chain RMSD values up to 3.01 Å). This showed that our approach is capable of sampling the conformational space of the residues in the binding pockets. Nevertheless, after equilibrium was reached all residues were settled at their most favorable conformation. To examine quantitatively the stability of the final conformations of the pseudo-peptides after equilibration, single-linkage-clustering was performed for all pivot residue structures

Table 3. Overall performance of the four prediction models using different data sets (ACC = accuracy (TP + TN)/(TP + TN + FP + FN), SEN = sensitivity (TP/TP + FN), SPC = specificity (TN/FP + TN), AUC = area under the curve (ROC analysis), TP = true positive, TN = true negative, FP = false positive, FN = false negative predictions).

Data set	MHCPEP				SYF + MHCPEP:NB				MHCPEP + SYF			
	ACC	SEN	SPC	AUC	ACC	SEN	SPC	AUC	ACC	SEN	SPC	AUC
SVMHC	0.78	0.81	0.76	0.86	0.81	0.84	0.79	0.90	0.83	0.91	0.73	0.88
YKW0201	0.82	0.89	0.76	0.88	0.81	0.68	0.89	0.91	0.84	0.89	0.76	0.89
DynaPred	0.77	0.77	0.78	0.87	0.78	0.67	0.84	0.85	0.79	0.88	0.66	0.85
$DynaPred^{POS}$	0.85	0.84	0.86	0.91	0.88	0.84	0.91	0.93	0.87	0.90	0.83	0.92

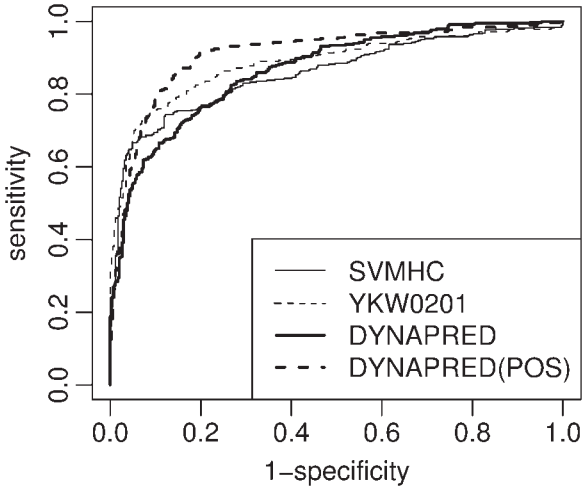


Fig. 2. ROC plots for overall performance evaluation using the MHCPEP data set.

sampled within the last 200ps of the simulations. Using a cutoff of 1.0 Å RMSD, only a single cluster was found for all pseudo-peptide simulations except for three cases. However, in these cases 194 to 199 structures out of 200 belonged to the first cluster and only 1–6 structures (0.5–3.0 %) were different. Since all additional clusters are under-represented, it is clear that the adoption of the corresponding conformations is only a rare event after the equilibrium is reached. Still, it shows that the MD approach is capable of sampling these conformations. Hence, we perceive that the average structure of the last 200ps represents the most favorable conformation of a bound pseudo-peptide in the binding pocket.

3.2 Prediction model

3.2.1 Overall performance To evaluate the performance of the models, we used the 10-fold cross validation or LOO (leave-one-out) techniques and calculated the Receiver Operating Characteristics Curve (ROC) (Sing *et al*., 2005). We compared our models to two models from the literature: the SVMHC model from (Dönnes *et al*., 2002) and the YKW0201 model from (Yu *et al*., 2002). We chose these two models for comparison, because in our method we use the quantitative-matrix approach combined with the SVM method. Thus it seemed sensible to compare our model to other methods using the same techniques, but no structural information. The SVMHC method uses SVM training of a simple binary vector approach, whereas the YKW0201 method uses a quantitative matrix, but no SVM. In addition, the YKW0201 model was previously compared to ANN and HMM methods and showed a comparable performance (Yu *et al*., 2002). Thus there was no need to include these methods into our comparison as well.

Table 3 and Fig. 2 depict the overall performance of the different methods obtained by LOO cross-validation. It can be observed that all methods perform well (>77% accuracy and >0.85 AUC). The SVMHC and YKW0201 methods show comparable performance on all data sets used. Because these methods are position-dependent—like all sequence based methods—they have to be compared to our position-dependent model. It can be observed that for all three data sets our position-dependent model, $DynaPred^{POS}$, outperforms all other models. The same can be seen in the ROC analysis as shown in Fig. 2. This shows that energetic data derived from structural studies are well suited as features for MHC-peptide binding prediction. The performance of the position-independent model, *DynaPred*, is only slightly lower than for the other three methods, despite the fact that no position information is included in this model. This shows that global structural features can be useful for binding prediction, although they do not perform as well as position-dependent models. Nevertheless, the position independent model is extremely robust with respect to the data set used (differs less than 1.5% ACC). The other methods show deviations of up to 5% accuracy (SVMHC) between different data sets.

3.2.2 Robustness Due to the high polymorphism of MHC molecules it is impossible to obtain large experimental binding data sets for all existing alleles. Thus it is important to test the performance of the methods with respect to their robustness against small data sets. In Fig. 3 the performance of the four models is given for different data set sizes. The results show that all methods except SVMHC have a comparably stable performance if the data set has more than 50 binders and 50 non-binders. On the contrary, the SVMHC model is highly dependent on the data volume, which is probably due to its simple binary encoding approach. Again the position-independent model shows the smallest variations above a data set size of 100 binders and 100 non-binders. Overall, the test shows that a data set with at least 100 binders and 100 non-binders is necessary for training a decent prediction model.

3.2.3 HIV-epitope prediction In the evaluation test on the HIV-genome the following accuracies were reached: SVMHC 82.72%,

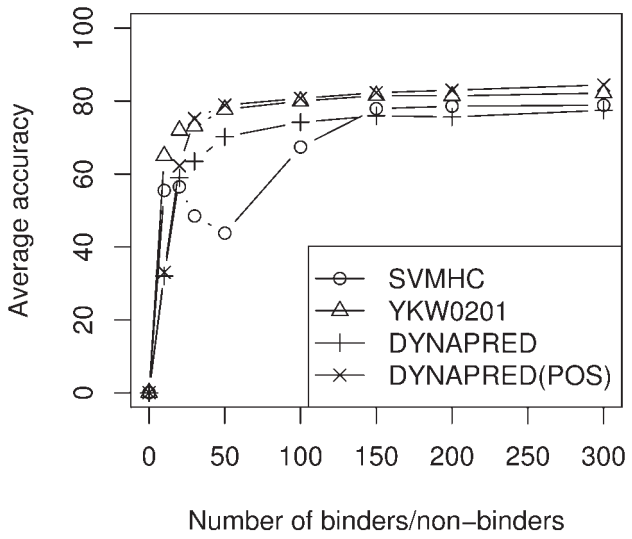


Fig. 3. Accuracy (%) of the 10-fold cross validation results for the training of the four models given in Table 3 using different data set sizes. The numbers correspond to the size of each of the two sets (binders (non-binders)).

YKW0201 82.11%, *DynaPred* 69.68%, *DynaPred*^{POS} 85.45%. Thus, the performance of the three position-dependent models is comparable to their performance on the training data set. The position-independent model shows a lower performance, which is surprising, because in all other tests it proved to be the more robust of our two models. Overall, the data shows that our models do perform nearly as well on independent data set as on the training data.

3.2.4 Generalizability The last test we performed on the four methods was a generalizability test on different HLA-A-type alleles. In Fig. 4 the percentage of correctly predicted binders by the models trained on the combined data set is given for the different alleles. It can be observed that in general the position-independent model considerably outperforms all other models, except for A2, which is the supertype of HLA-A*0201, and thus contains mainly HLA-A*0201 sequences. The prediction capabilities of SVMHC, YKW0201, and our position-dependent model are mostly between 10–30% implying that cross-allele prediction is not feasible for them.

3.3 Construction of peptide conformations

The last step of our prediction algorithm is the construction of bound peptide conformations for all predicted binding sequences. For testing this step, we generated bound conformations for all peptide sequences of the structures given in Table 4 by connecting the saved residue conformations from the simulation runs and performing a short energy minimization. At this point we abstained specifically from further structural refinement, because we wanted to evaluate two points crucial for our method: First, are we able to construct a decent peptide backbone structure by simply 'stitching' together the pivot residue conformations and subsequent energy minimization. This was not obvious at the beginning, because the pseudo-peptide backbone was still able to move within a few Å even with the restraints applied. Second, if we were able to do so, is the overall energy of the constructed peptides comparable to the energies of the experimental peptide structures. This would be a prove for the validity of our additive single residue approach and in addition, is a prerequisite for a possible use of the constructed peptides for further refinement and the calculation of binding affinities from the complex structures.

For this evaluation, we calculated the backbone RMSD and the differences in the potential energies between the constructed peptides in the binding groove and the peptides in the experimental structures. The RMSD values are given in Table 4. RMSD^{1AKJ} provides a measure for the difference between the backbones of the experimental structures and the backbone of 1AKJ. RMSD^{Gen} compares the backbone of the constructed peptides to the crystal structures. Comparing the data shows that the deviation of our constructed backbone structures from the experimental structures is comparable to the variation between the experimental structures. This proves that even with this rather simple approach we can generate decent backbone peptide structures based on our residue conformations.

To investigate the correlation between the energies of the constructed and experimental structures, we compared the potential energies of the bound peptide structures for both sets. The correlation plot is shown in Fig. 5. A correlation of 0.81 was found between the energies, validating that the energies derived from our single residue conformations are suited for prediction and ranking purposes. However, a general energy setoff can be observed in the plot. This is due to two reasons: First, a rather high average RMSD value (3.8 Å, data not shown) for the solvent exposed side chains was observed. The treatment of these residues posed also a problem for all previously reported structural studies, due to the lack of solvent. Thus in most studies the RMSD of the solvent exposed residues is either in the same range as reported here or these residues are placed according to known X-ray structure conformations. Second, during the minimization of the peptide backbone, the side chain conformations are distorted. Due to the simple refinement strategy used, the side chains might not re-equilibrate into their global, minimum conformation, but rather a local minimum. However, the average RMSD for the buried anchor side chains is only 1.1 Å. Thus, both RMSD values—backbone and buried side chains—are in the same range as in other docking approaches.

4 DISCUSSION

We present a new combined structure- and sequence-based method for the prediction of MHC-binding peptides, in which residue-based energy terms from MD simulations are used as features to train a position-dependent (*DynaPred*^{POS}) and a position-independent (*DynaPred*) prediction model for peptide-MHC class I binding using SVMs. The performance of the prediction models was tested successfully on the HIV genome as an independent test set. Our position-dependent model outperforms the two other sequence-based models in our evaluation, validating that structure based energies are well suited as features for binding prediction. The position-independent model showed a lower performance (∼5% accuracy) than the position-dependent models, but had a much higher generalizability towards other HLA-A-type alleles. This is in agreement with the performance of other prediction models based only on global features (Doytchinova *et al*., 2005; Bordner and Abagyan, 2006). The high generalizability of methods based on global features can be explained by the fact that for HLA-subtypes

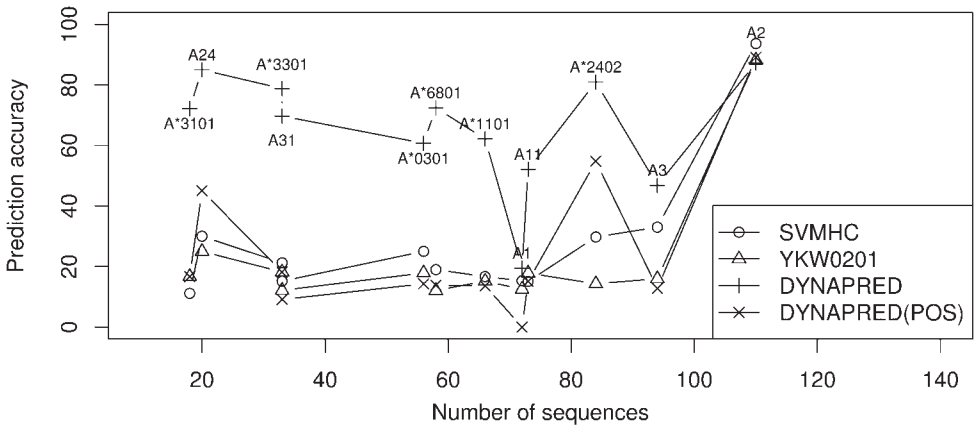


Fig. 4. Correctly predicted binders (%) by the four prediction models from Table 3 trained on the combined data set (MHCPEP + SYF) on various HLA-A-type alleles. The alleles are ordered according to the number of peptide sequences available for the specific allele.

Table 4. Backbone-RMSD (Å) between the generated peptides and the crystal structures ($RMSD^{Gen}$) and between the experimental structures and 1AKJ ($RMSD^{1AKJ}$).

PDB	Resolution	Sequences	$RMSD^{1AKJ}$	$RMSD^{Gen}$
1AKJ	2.65	ILKEPVHGV	0.00	1.18
1AO7	2.60	LLFGYPVYV	1.22	1.58
1B0G	2.50	ALWGFFPVL	1.23	1.41
1BD2	2.50	LLFGYPVYV	1.24	1.59
1DUZ	1.80	LLFGYPVYV	1.33	1.68
1HHG	2.60	TLTSCNTSV	1.67	1.38
1HHI	2.50	GILGFVFTL	1.38	1.67
1HHJ	2.50	ILKEPVHGV	0.52	1.29
1HHK	2.50	LLFGYPVYV	1.29	1.69
1I1F	2.80	FLKEPVHGV	0.50	1.32
1I1Y	2.20	YLKEPVHGV	0.66	1.41
1I7R	2.20	FAPGFFPYL	1.32	1.57
1I7T	2.80	ALWGVFPVL	1.17	1.62
1I7U	1.80	ALWGVFPVL	1.26	1.76
1IM3	2.20	LLFGYPVYV	1.29	1.61
1JHT	2.15	ALGIGILTV	1.46	1.32
1OGA	1.40	GILGFVFTL	1.62	1.67
1QRN	2.80	LLFGYAVYV	1.29	1.84
1QSE	2.80	LLFGYPRYV	1.17	1.33
1QSF	2.80	LLFGYPVAV	1.10	1.63
Average RMSD			1.14	1.53

often only one or two of the nine binding pockets have different binding site residues. These local differences do strongly affect position-dependent methods, but are averaged out by the use of global features. The generalizability of prediction methods is highly desirable because of the polymorphism of the MHC molecules and the need of 'supertype' MHC binders for purposes like vaccine design. In addition, there is still a severe lack of experimental binding data for less common HLA-types, thus preventing the training of prediction models for these types. This makes highly generalizable models, which also work for these HLA-types, an alternative. However, the generalizability comes at the price of lower accuracy (about 5% less). Thus our prediction approach,

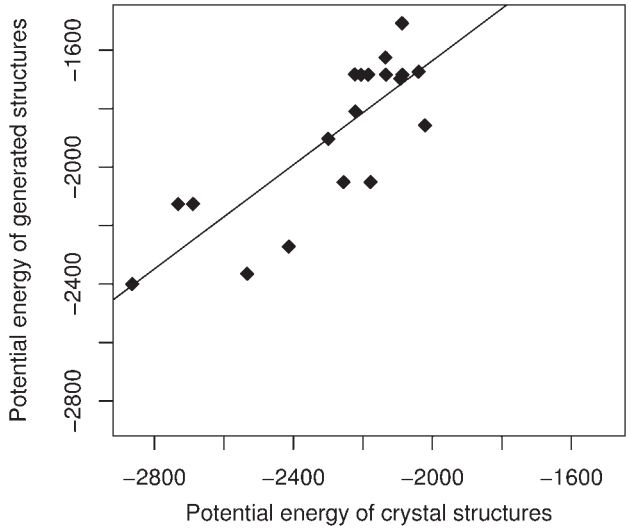


Fig. 5. Correlation between the potential energy of the constructed structures and the crystal structures (kJ/mol).

which uses the same features for position-dependent and position-independent prediction models, and thus allows using either model depending on the allele and purpose of the study, may be an attractive choice.

We showed that with our molecular-dynamics-based approach it is possible to sample the residues conformational space within each binding pocket adequately. Based on these simulations, we are able to construct decent conformations of bound peptides, which have RMSD values that are comparable with the results of other docking studies. In addition, the correlation between the potential energies of the constructed peptides and the potential energies for the corresponding experimental structures is as high as 0.81. This validates the use of our equilibrated residue structures for prediction as well as for peptide construction and is in agreement with a former study, in which a linear energy approach based on MD simulations performed very well for the calculation of free energies of binding of small peptides to OppA (Wang *et al.*, 2002). The low RMSD values and high energy correlation obtained

for the constructed peptides are very promising, especially considering that only a simple approach of concatenation and energy minimization was used. Due to the pre-calculation of the residue structures, the concatenation and minimization is extremely fast, compared with other docking methods. Thus our method provides a fast alternative to generate the initial docked structure which can be refined subsequently for binding affinity prediction. However, the efficiency of our method comes at a price, which is the necessity to pre-calculate the bound conformations of the single residues for each binding pocket. Thus to use our method for other protein targets these conformations must first be calculated for this target. This distinguishes our method considerably from other docking methods such as Liu *et al*. 2004 and Bordner *et al*. 2006. However, the purpose of this work was not to develop a general protein-peptide docking method, but to improve MHC/peptide binding prediction by the use of structural features. For this purpose the higher compute-time efficiency of our approach is more important than transferability. In addition, it is still an open question to what extend new simulations need to be performed to compute bound peptide conformations for other MHC alleles, especially if the allele-specific binding pocket mutations are conservative or only one or two side chains differ in the binding pocket. There are several other interesting topics to be investigated in the future: For example, like all other structural approaches, we are experiencing problems with the treatment of the solvent-exposed residues. To solve this problem, further refinement strategies should be investigated. In addition, the performance of our method for regression should be evaluated and it would be interesting to try to further improve the accuracy of the structure based position-independent model.

ACKNOWLEDGEMENTS

We thank V. Brusic for providing the non-binding data of the MHCPEP data base, K. Roomp and J. Rahnenführer for many inspiring and helpful discussions, and J. Büch for technical support. Funding was obtained from the IMPRS program of the Max-Planck-Society.

REFERENCES

Aqvist,J., Luzhkov,V.B. and Brandsdal,B.O. (2002) Ligand Binding Affinities from MD Simulations. *Acc. Chem. Res.,* **35**, 358–365.

Altuvia,Y., Schueler,O. and Margalit,H. (1995) Ranking potential binding peptides to MHC molecules by a computational threading approach. *J. Mol. Biol.* **249**, 244–250.

Bhasin,M. and Raghava,G.P.S. (2004) Prediction of CTL epitopes using QM, SVM and ANN techniques. *Vaccine,* **22**, 3195–3204.

Bordner,A.J. and Abagyan,R. (2006) Ab initio prediction of peptide-MHC binding geometry for diverse class I MHC allotypes. *Proteins*, in press.

Brusic,V., Rudy,G., Kyne,A.P. and Harrison,L. (1998a) MHCPEP, a database of MHC-binding peptides (updated 1997). *Nucl. Acid. Res.,* **26**, 368–371.

Brusic,V., Rudy,G., Honeyman,M., Hammer,J. and Harrison,L. (1998b) Prediction of MHC Class II binding peptides using an evolutionary algorithm and artificial neural network. *Bioinformatics,* **14**, 121–130.

Bui,H.H., Sidney,J., Peters,B.M., Sinichi,A., Purton,K.A., Mothé,B.R., Chisari,F.V., Watkins,D.I. and Sette,A. (2005) Automated generation and evaluation of specific MHC binding predictive tools: ARB matrix applications. *Immunogenetics,* **57**, 304–314.

Canutescu,A.A., Shelenkov,A.A. and Dunbrack,R.L.,Jr. (2003) A graph theory algorithm for protein side-chain prediction. *Protein Science,* **12**, 2001–2014.

Chothia,C. (1974) Hydrophobic bonding and accessible surface area in proteins. *Nature,* **248**, 338–339.

Desmet,J., Wilson,I.A., Joniau,M., Maeyer,M. and Lasters,I. (1997) Computation of the binding of fully flexible peptides to proteins with flexible side chains. *FASEB J.,* **11**, 164–172.

Doytchinova,I.A., Blythe,M.J. and Flower,D.R. (2002) Additive method for the prediction of protein-peptide binding affinity: Application to the MHC class I molecule HLA-A*0201. *J Proteome Res,* **1**(3), 263–272.

Doytchinova,I.A., Guan,P. and Flower,D.R. (2004) Quantitative structure-activity relationships and the prediction of MHC supermotifs. *Methods*, **34**, 444–453.

Doytchinova,I.A., Walshe,V., Borrow,P. and Flower,D.R. (2005) Towards the chemometric dissection of peptide—HLA-A*0201 binding affinity: Comparison of local and global QSAR models. *J. Comp-Aided Mol. Design,* **19**, 203–212.

Dönnes,P. and Elofsson,A. (2002) Prediction of MHC class I binding peptides using SVMHC. *BMC Bioinformatics,* **3**, 25.

Dunbrack,R.L., Canutescu,A.A. and Shelenkov,A.A. (2003) A graph theory algorithm for protein side-chain prediction. *Prot. Sci.* **12**, 2001–2014.

Gulukota,K., Sidney,J., Sette,A. and DeLisi,C. (1997) Two complementary methods for predicting peptides binding major histocompatibility complex molecules. *J. Mol. Biol.,* **267**, 1258–1267.

Hammer J. (1995) New methods to predict MHC-binding sequences within protein antigens. *Curr. Opin. Immunol.,* **7**, 263–269.

Korber,B.T.M., Brander,C., Haynes,B.F., Koup,R., Moore,J.P., Walker,B.D. and Watkins,D.I. (ed.) (2005) *HIV Molecular Immunology 2005*, Los Alamos National Laboratory, Theoretical Biology and Biophysics, Los Alamos, New Mexico. LA-UR 06-0036.

Lindahl,E., Hess,B. and Spoel,D. (2001) GROMACS 3.0: A package for molecular simulation and trajectory analysis. *J. Mol. Modelling* **7**, 306–317.

Liu,Z., Dominy,B.N. and Shakhnovich,E.I. (2004) Structural Mining: Self-Consistent Design on Flexible Protein-Peptide Docking and Transferable Binding Affinity Potential. *J. Am. Chem. Soc.,* **126**(27), 8515–8528.

Logean,A. and Rognan,D. (2002) Recovery of known T-cell epitopes by computational scanning of a viral genome. *J. Comp-Aided Mol. Design,* **16**, 229–243.

Mamitsuka,H. (1998) Predicting peptides that bind to MHC molecules using supervised learning of Hidden Markov Models. *Proteins: Structure, Function and Genetics,* **33**, 460–474.

Milik,M., Sauer,D., Brunmark,A.P., Yuan,L., Vitiello,A., Jackson,M.R., Peterson,P.A., Skolnick,J. and Glass,C.A. (1998) Application of an artificial neural network to predict specific class I MHC binding peptide sequences. *Nat. Biotech.,* **16**(8): 753–756.

Nielsen,M., Lundegaard,C., Worning,P., Hvid,C.S., Lamberth,K., Buus,S., Brunak,S. and Lund,O. (2004) Improved prediction of MHC class I and class II epitopes using a novel Gibbs sampling approach. *Bioinformatics,* **20**(9), 1388–1397.

Ota,N. and Agard,DA. (2001) Binding mode prediction for a flexible ligand in a flexible pocket using multi-conformation simulated annealing pseudo crystallographic refinement. *J. Mol. Biol.,* **314**, 607–617.

Parker,K.C., Bednarek,M.A. and Coligan,J.E. (1994) Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side chains. *J. Immunol.,* **152**, 163–175.

Paul,W.E. (ed.) (1998) Fundamental Immunology, 4th Edn. Raven Press, New York, NY.

Peters,B., Tong,W., Sidney,J., Sette,A. and Weng,Z. (2003) Examining the independent binding assumption for binding of peptide epitopes to MHC-I molecules. *Bioinformatics,* **19**(14), 1765–1772.

Pickett,S.D. and Sternberg,M.J. (1993) Empirical scale of side-chain conformational entropy in protein folding. *J. Mol. Biol.,* **231**, 825–839.

Rammensee,H.G., Bachman,J., Philipp,N., Emmerich,N., Bachor,O.A. and Stevanovic,S. (1999) SYFPEITHI: a database for MHC ligands and peptide motifs. *Immunogenetics,* **50**, 3–9.

Devel R. Core Team. (2005) R: A Language and Environment for Statistical Computing. Vienna, Austria. http://www.R-project.org.

Reche,P.A., Glutting,J.P. and Reinherz,E.L. (2002) Prediction of MHC class I binding peptides using profile motifs. *Hum. Immunol.,* **63**, 701–709.

Rosenfeld,R., Zheng,Q., Vajda,S. and DeLisi,C. (1993) Computing the structure of bound peptides—Application to antigen recognition by class I major histocompatibility complex receptors. *J. Mol. Biol.,* **234**, 515–521.

Schueler-Furman,O., Elber,R. and Margalit,H. (1997) Knowledge-based structure prediction of MHC class I bound peptides: A study of 23 complexes. *Fold. Des.* **3**, 549–564.

Segal,M.R., Cummings,M.P. and Hubbard,A.E. (2001) Relating amino acid sequence to phenotype: Analysis of peptide-binding data. *Biometrics,* **57**, 632–642,

Sette,A., Buus,S., Appella,E., Smith,J.A., Chesnut,R., Miles C.,Colon SM. and Grey HM. (1989) Prediction of major histocompatibility complex binding regions

of protein antigens by sequence pattern analysis. *Proc. Natl Acad. Sci. USA,* **86**, 3296.

Sezerman,U., Vajda,S. and DeLisi,C. (1996) Free energy mapping of class I MHC molecules and structural determination of bound peptides. *Prot. Sci.,* **5**, 1272–1281.

Sing,T., Sander,O., Beerenwinkel,N, and Lengauer,T. (2005) ROCR: visualizing classifier performance in R, *Bioinformatics,* **21**(20), 3940–3941.

Southwood,S., Sidney,J., Kondo,A., Guercio,M., Appella,E., Hoffman,S., Kubo,R.T., Chesnut,R.W., Grey,H.M. and Sette,A. (1998) Several common HLA-DR types share largely overlapping peptide binding repertoires. *J. Immunol.,* **160**, 3363–3373.

Ting,W. and Wade,R.C. (2002) Comparative Binding Energy (COMBINE) Analysis of OppA-Peptide Complexes Relate Structure to Binding Thermodynamics. *J. Med. Chem.,* **45**, 4828–4837.

Tong,J.C., Tan,T.W. and Ranganathan,S. (2004) Modeling the structure of bound peptide ligands to Major Histocompability Complex. *Prot. Sci.,* **13**, 2523–2532.

Yang Z.R. and Johnson, F.C. (2005) Prediction of T-Cell Epitopes Using Biosupport Vector Machines. *J. Chem. Inf. Model.,* **45**, 1424–1428.

Yu,K., Petrovsky,N., Schonbach,C., Koh,J.Y. and Brusic, V. (2002) Methods for prediction of peptide binding to MHC molecules: A comparative study. *Molecular Medicine*, **8(3)**, 137–148.

Zhao,Y., Pinilla,C., Valmori,D., Martin,R. and Simon,R. (2003) Application of support vector machines for T-cell epitopes prediction. *Bioinformatics,* **19**(15), 1978–1984.

Zhihua,L., Yuzhang,W., Bo, Z., Bing,N. and Li,W. (2004) Toward the Quantitative Prediction of T-Cell Epitopes: QSAR Studies on Peptides Having Affinity with the Class I MHC Molecular HLA-A*0201. *J. Comp. Biol.,* **11**(4), 683–694.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e25–e34
doi:10.1093/bioinformatics/btl244

ProfilePSTMM: capturing tree-structure motifs in carbohydrate sugar chains

Kiyoko F. Aoki-Kinoshita*,†, Nobuhisa Ueda, Hiroshi Mamitsuka and Minoru Kanehisa

Bioinformatics Center, Institute for Chemical Research, Kyoto University

ABSTRACT

Motivation: Carbohydrate sugar chains, or glycans, are considered the third major class of biomolecules after DNA and proteins. They consist of branching monosaccharides, starting from a single monosaccharide. They are extremely vital to the development and functioning of multicellular organisms because they are recognized by various proteins to allow them to perform specific functions. Our motivation is to study this recognition mechanism using informatics techniques from the data available. Previously, we introduced a probabilistic sibling-dependent tree Markov model (PSTMM), which we showed could be efficiently trained on sibling-dependent tree structures and return the most likely state paths. However, it had some limitations in that the extra dependency between siblings caused overfitting problems. The retrieval of the patterns from the trained model also involved manually extracting the patterns from the most likely state paths. Thus we introduce a profilePSTMM model which avoids these problems, incorporating a novel concept of different types of state transitions to handle parent-child and sibling dependencies differently.

Results: Our new algorithms are more efficient and able to extract the patterns more easily. We tested the profilePSTMM model on both synthetic (controlled) data as well as glycan data from the KEGG GLYCAN database. Additionally, we tested it on glycans which are known to be recognized and bound to proteins at various binding affinities, and we show that our results correlate with results published in the literature.

Contact: kkiyoko@t.soka.ac.jp

1 INTRODUCTION

Carbohydrate sugar chains, or glycans, are considered the third major class of biomolecules after DNA and proteins. They consist of branching monosaccharides, starting from a single monosaccharide, usually bound to a protein on the cell surface. They are extremely vital to the development and functioning of multicellular organisms because they are recognized by various proteins to allow them to perform specific functions. Oftentimes, these functions change depending on the different glycans that are bound to the protein.

Although these glycans are known to be vital, due to their structural complexity, they are not as well understood as DNA or protein sequences. Within the last few years, however, a major movement to advance bioinformatics for glycans has been underway. Thanks to the data left by the CarbBank project [8], resources such as KEGG (Kyoto Encyclopedia of Genes and Genomes) [14,18], CFG (Consortium for Functional Glycomics), glycosciences.de [20], and now the EuroCarbDB resource, have been able to compile information on glycans quickly. These resources enable informatics techniques to be directly applied to glycan data to aid researchers to better understand the functions and structures of these complicated molecules. In the past couple of years, this field which we dub *glycome informatics* has taken off, with the development of glycan structure comparison [3] and score matrix [1] algorithms, a Composite Structure Map (CSM) [15] for delineating all possible carbohydrate structures, and mass spectra prediction algorithms [11].

It is generally understood that glycans are recognized by various proteins (lectins), which allow them to take on a variety of functions. This recognition mechanism is still currently being investigated by many glycobiologists [17,19,24] for various carbohydrate-binding proteins. Our aim is to study this mechanism using informatics techniques from the data available. Towards this aim, we presented our first work on capturing patterns in glycan structure data in the form of a probabilistic model containing sibling-dependencies, called PSTMM for probabilistic sibling-dependent tree Markov model [2,22]. We added an additional dependency to the hidden tree Markov model [7] between consecutive siblings in order to capture the ordering of children, and we were able to develop sufficiently efficient algorithms to train this model.

PSTMM utilized a set of states that output a distribution of a set of labels, where any state could transition to any other state. This provided flexibility such that any pattern in any arrangement could be learned. However, there were several drawbacks. First, by allowing all states to transition between all states, the computation time was cubic on the order of states. Although this was still within the practical maximal bounds for a probabilistic model, it is still rather expensive. Second, although an algorithm to extract the most likely state paths was provided, we still required the manual extraction of patterns from these paths. That is, we were left with the most likely paths, but the interpretation of what patterns from the data these corresponded to required some manual efforts. Third, the increased dependency between siblings added the risk of overfitting to the data unless sufficiently large amounts of data were examined.

In this work, we introduce a model that overcomes these drawbacks. Considering how profile hidden Markov models [10] improved on hidden Markov models [9] simply by incorporating new types of states whose positions were fixed, we could consider a similar improvement to PSTMM. However, this is insufficient (and uninteresting) as a new model. Our new model is in fact significantly different because not only did we add new types of states, we needed to consider the sibling relationships and parent-child

*To whom correspondence should be addressed.
†Currently at Department of Bioinformatics, Faculty of Engineering, Soka University

relationships differently. We needed to be able to distinguish between these two types of transitions in this new model because of the fixed positions of the states. This novel technique of utilizing different types of state transitions completed our new model. We could then replace the original algorithms with new ones, making them more efficient and able to extract patterns more easily. Because the states are fixed in this model, it can also avoid overfitting problems that may occur when many dependencies exist.

Here, we present this new model called profilePSTMM and provide the new algorithms used to train it. We also tested this model on both synthetic (controlled) data as well as glycan data from the KEGG GLYCAN database. Additionally, we tested it on glycans which are known to be recognized and bound to proteins at various binding affinities, and we show that our results correlate with results published in the literature. Furthermore, we evaluated how well profilePSTMM can distinguish between different classes of glycans. We discuss these drastic improvements in performance. Finally, we discuss how this new model may be applied to other problems in glycobiology.

2 BACKGROUND

Before introducing our new model, it is necessary to clarify the motivation behind our work as well as the notation used in our model. So we will briefly describe glycan structures and our previous PSTMM model in this section.

2.1 Notation and terminology

The following terminology will be used throughout this paper. We refer to a *tree* as an acyclic connected graph with vertices of the tree defined as *nodes*. A tree is *rooted* if it branches off from a single node, called the *root*. Any node x on a unique path from the root to y is called the *ancestor* of y, making y a descendant of x. Any descendant y that is connected to x by a single edge is a *child* of x, making x the *parent* of y. Children of the same parent are *siblings* and a node with no children is a *leaf*. A *subtree* of tree T is a tree whose nodes and edges are a connected subset of T, and an *ordered tree* is the rooted tree where the children of each node are ordered. A *labeled tree* is a tree in which a label is attached to each node. All trees in this paper are considered ordered, labeled and rooted trees. The *level* of a node in a tree is defined as the distance of the node from the root. Thus the root is at level 0, its children are at level 1, whose children are at level 2, and so on.

The following notation for equations given later to describe our model will be used. Let $\mathbf{T} = \{T_1, \ldots, T_{|\mathbf{T}|}\}$ be a set of labeled ordered trees, where $T_u = (V_u, E_u)$, $V_u(=\{x_1^u, \ldots, x_{|V_u|}^u\})$ is a set of nodes, and E_u is a set of edges. For a node x_i, we may simply use the notation i when it is clear from the context. x_1^u is the root of tree T_u, $|V| = \max_u|V_u|$, $t_u(i)$ is a subtree of T_u, having x_i^u as the root of $t_u(i)$, and $C_u(p) \subseteq \{1, \ldots, |C_u(p)|\}$ is a set of indices of children of x_p^u in T_u. Let $|C| = \max_{u,p}|C_u(p)|$. Let $x_{\leftarrow}^u(p)$ and $x_{\rightarrow}^u(p)$ be the eldest and youngest child of node p, respectively. Each node x_j^u has label $o_j^u \in \Sigma$, where $\Sigma = \{\sigma_1, \ldots, \sigma_{|\Sigma|}\}$ is the set of labels (i.e., the alphabet) applied to the nodes. For node j, we will use i, k and p to refer to the immediately elder sibling, the immediately younger sibling, and the parent, respectively. Also note that the superscript u in our notations (such as node x^u and label o_j^u) referring to a variable in tree u will often be ommitted in the text when understood from the context.

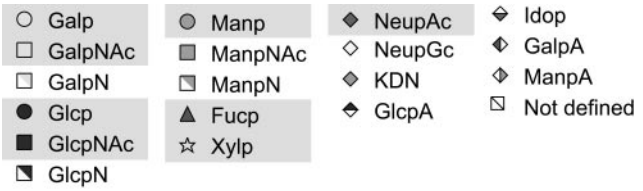


Fig. 1. A list of the standard symbols for monosaccharides, as defined by the CFG (see text for details).

2.2 Glycans and glycobiology

A basic overview of glycobiology can be found in a book by Varki *et al*. [24], so we will only review the basic structures and classes of glycans which we refer to in this work.

2.2.1 Glycan structures The Consortium for Functional Glycomics (CFG) is an international consortium of research institutes and universities worldwide, focusing on providing a central and freely available resource of glycan-related data including mass spectroscopy data and glycan array expression data. The CFG has established a standard notation for common monosaccharides, as given in Figure 1. We will be using this notation in the text.

Each monosaccharide is connected to one or more monosaccharides, forming a branched structure that as a whole is considered a glycan. Glycans are usually drawn from right to left, with the root located at the right, and children branching out to the left. Thus when referring to linkages in a glycan drawn in this way, we will specify them from left-to-right (as if reading them in English), towards the root. Monosaccharides are linked to one another in various conformations, indicated by the anomer (α or β) and hydroxyl group to which they are linked. Oftentimes these detailed conformations are unknown, thus necessitating a probabilistic model for capturing patterns as opposed to algorithms that require the details to be known in advance. Note that in this paper, since sequences are usually read left to right, we will draw our model from top-down with the root at the top so that siblings can be read left-to-right, while glycans and profiles will be drawn from right to left.

2.2.2 Glycan classes Glycans are currently classified according to their *core structure*, which is the subset of common structures around the root monosaccharide. The most commonly studied class is the N-Glycan class, which is characterized by a $Manp_3$-$GlcpNAc_2$ structure as its core. The O-Glycan class is characterized by a smaller core structure which is also subdivided into several subtypes.

The list of glycan classes and their sizes are given in Table 1. We make note here that classifications are mainly determined manually by an expert, especially for those that do not involve a core structure. Thus, these classifications are not ''perfect'' meaning that there may be many discrepancies due to human error. The classifications in KEGG GLYCAN are also hierarchical, so for example GPI anchors are a subclass of Glycoproteins. However not all Glycoproteins could be subclassified, so the most detailed class names were counted, and some glycans can also be classified into more than one class, so they are multiply-counted in this table.

2.3 PSTMM

The probabilistic sibling-dependent tree Markov model was shown to be able to capture patterns in tree structures, especially glycans

Table 1. Glycan classes in KEGG Glycan as of Feb. 2, 2006. Only the lowest level class names were used to calculate the statistics

Class	Total Num.	Avg. Num. Nodes
N-Glycans	2173	11.0
Sphingolipid	931	6.8
Glycoside	878	3.4
O-Glycan	778	6.1
LPS	767	6.7
Glycosaminoglycan	580	7.6
Polysaccharide	473	6.6
Glycolipid	163	4.0
GPI anchor	134	6.4
Glycoprotein	106	4.4
Oligosaccharide	77	5.1
Neoglycoconjugate	54	5.1
Glycerolipid	20	3.6

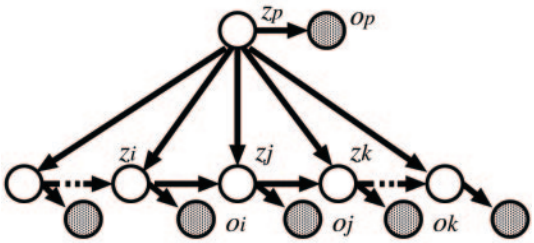


Fig. 2. PSTMM model where dependencies exist between consecutive siblings.

[2,22]. Algorithms were also developed which could estimate the parameters and find the most likely paths within the practical bounds of the maximum known limits. In comparison to tree Markov models, PSTMM included dependencies between siblings such that the order between them could be maintained, as illustrated in Figure 2. In addition to the classic forward and backward parameters of Baum-Welch, upward and downward parameters were incorporated to efficiently estimate the parameters. A tree's parameters would be estimated starting from the leaves and traveling up the parents, and forward and backward between siblings, up to the root. Then the downward parameter would be estimated in a breadth-first fashion from the root going back down. These four parameters were used to calculate the expectation values for the state transition probability, output label probability and initial state probability values. The maximum likelihood value would then be estimated, the probability parameters updated using the expectation values, and the process would be repeated until the maximum likelihood converged. Finally, most likely state paths could be estimated by finding the states providing the highest probability values.

Experiments using PSTMM were performed on both synthetically generated tree structures and glycan structures, and it was shown that patterns could indeed be captured better than previous models. In fact, the model was trained on the most popular classes of glycans called N-Glycans, and PSTMM found the three known subclasses of N-Glycans: hybrid, high-mannose, and complex type, which are characterized by patterns at their leaves. Thus the utility of this model in bioinformatics was illustrated.

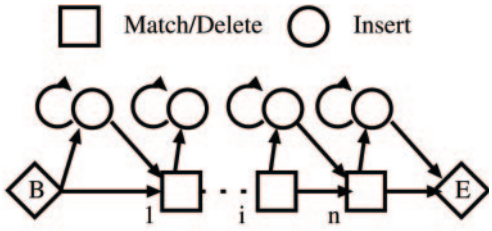


Fig. 3. ProfileHMM structure where match and delete states are combined.

3 METHOD

Although PSTMM could find the subclasses of N-Glycans from within the data, it was a tenuous procedure to extract these patterns. Additionally, the algorithm to estimate the parameters iterated through all possible state transitions, which resulted in very long computation times especially as more states were added. Thus, we rebuilt the model into a new model which we introduce in this section.

3.1 ProfilePSTMM structure

In order to describe the structure of profilePSTMM, it would help to describe the simpler profileHMM structure first. Figure 3 illustrates the profileHMM structure using our notation combining the match and delete states at the same positions. There is a Begin and End state from which the model begins and ends, respectively. Insert states loop back to themselves to handle consecutive gaps in the sequence.

Our new model called profilePSTMM also incorporates new insert and delete states in addition to the existing match states, whose positions are fixed in the state model. These three states make up a set, which is fixed at a specific position in the state model. We use M_i, I_i, and X_i to indicate match, insert and delete states, respectively, at position i. The challenge that we were then faced with was how to distinguish between transitions from parent to child and between siblings between the fixed positions. So we came upon the idea to introduce different types of state transitions. Figure 4 illustrates our new model. These new state transitions are called *Down* for parent-child transitions and *Right* for sibling-sibling transitions. We can consider the *Right* transitions as the siblings of one family, corresponding to one profileHMM. When a child node i is not a leaf, when $C(i) > 0$, it would have *Down* transitions as if it were state q in the figure. These state transitions are differentiated in the figure according to color, and the black lines indicate that both transitions occur between the indicated states. A Begin state transitions down to the root node n_1. In fact, the Begin state also serves as an End state in our model since the parameters are calculated and accumulated there.

3.2 Parameters and auxiliary probabilities

ProfilePSTMM has three probability parameters, π, a and b. The initial state probability $\pi[s_l]$ $(= P(z_1^u = s_l; \theta))$ is the probability that state (z_1^u) of root node x_1^u is s_l, the state transition probability $a[\{s_q, s_l\}, s_m]$ $(= P(z_j^u = s_m | z_p^u = s_q, z_i^u = s_l; \theta))$ is the conditional probability that the state of a node x_j^u is s_m given that the states of its parent (x_p^u) and immediately elder sibling (x_i^u) are s_q and s_l, respectively, and the label output probability $b[s_l, \sigma_h]$ $(= P(o_j^u = \sigma_h | z_j^u = s_l; \theta))$ is the conditional probability that the output label of node x_j^u is σ_h given that the state of x_j^u is s_l.

These probability parameters are estimated using the same forward, backward, upward and downward probabilities as PSTMM, except now taking into consideration the state position and the different types of states and state transitions. The forward probability $F_j(s_q, s_l)$ is the probability that for node j, all labels of the subtrees of each of the elder siblings are generated, the state of node j is s_l, and the state of parent p is s_q. The following forward probability equations are now defined as follows

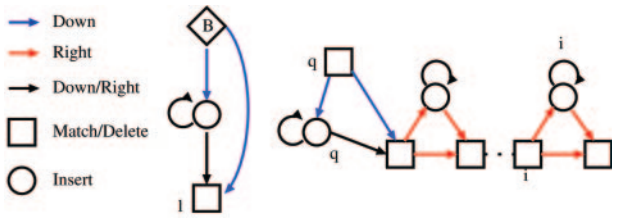


Fig. 4. New profilePSTMM state model with match, insert and delete states. New state transitions are called *Down* for parent-child transitions and *Right* for sibling-sibling transitions. These state transitions are differentiated according to blue and red color, and the black lines indicate that both transitions occur between the indicated states. Because match and delete states are always found together, they have also been combined for clarity. This figure represents just one downward step for one parent and its children.

depending on the state type.

$$F_j(s_q, M_l) = \begin{cases} \text{If } x_j = x_{\leftarrow}(p) \text{ then } a[\{s_q, -\}, M_l], \\ \text{o.w.} \\ \quad F_i(s_q, M_k) U_i(M_k) a[\{s_q, M_k\}, M_l] + \\ \quad F_i(s_q, I_k) U_i(I_k) a[\{s_q, I_k\}, M_l] + \\ \quad F_i(s_q, X_k) U_i(X_k) a[\{s_q, X_k\}, M_l] \end{cases}$$

where x_i is the older brother of x_j and s_k is the state of x_i.When s_l is an insert state, we need to take into consideration the self-loop. Thus the formula becomes

$$F_j(s_q, I_l) = \begin{cases} \text{If } x_j = x_{\leftarrow}(p) \text{ then } a[\{s_q, -\}, I_l], \\ \text{o.w.} \\ \quad F_i(s_q, M_l) U_i(M_l) a[\{s_q, M_l\}, I_l] + \\ \quad F_i(s_q, I_l) U_i(I_l) a[\{s_q, I_l\}, I_l] + \\ \quad F_i(s_q, X_l) U_i(X_l) a[\{s_q, X_l\}, I_l] \end{cases}$$

where x_i is the older brother of x_j. The forward parameter when s_l is a delete state is the same as for when it is a match state.

The backward probability $B_j(s_q, s_m)$ is the probability that for node j, all labels of the subtrees of each of the younger siblings and node j are generated, s_m is the state of j, and s_q is the state of its parent. For the backward probability, the same equation can be used for any type of state s_k, as follows:

$$B_i(s_q, s_k) = \begin{cases} \text{If } x_i^u = x_{\rightarrow}^u(p) \text{ then } U_i(s_k), \\ \text{o.w.} \\ \quad U_i(M_k) a[\{s_q, s_k\}, M_l] B_j(s_q, M_l) + \\ \quad U_i(I_k) a[\{s_q, s_k\}, I_k] B_j(s_q, I_k) + \\ \quad U_i(X_k) a[\{s_q, s_k\}, X_l] B_j(s_q, X_l)) \end{cases}$$

where x_j is the younger brother of x_i and s_l is the state of x_j.

The upward probability $U_p(s_q)$ is the probability that all labels of subtree $t(p)$ are generated and that the state of node p is s_q. The upward probability is also different for different state types. Here we combined the different options into a single equation. The label output probability when state s_q is a delete state is set to 1 (0 in log values).

$$U_p(s_q) = \begin{cases} \text{If } C_u(p) = \emptyset \text{ then} \\ \quad \text{if } s_q \text{ is a delete state then } 1 \\ \quad \text{else } b[s_q, o_p], \\ \text{o.w.} \\ \quad \text{if } s_q \text{ is a match or insert state then} \\ \quad b[s_q, o_p](F_j(s_q, M_m) B_j(s_q, M_m) + \\ \qquad F_j(s_q, I_m) B_j(s_q, I_m) + \\ \qquad F_j(s_q, X_m) B_j(s_q, X_m)) \\ \quad \text{else if } s_q \text{ is a delete state then} \\ \qquad (F_j(s_q, M_m) B_j(s_q, M_m) + \\ \qquad F_j(s_q, I_m) B_j(s_q, I_m) + \\ \qquad F_j(s_q, X_m) B_j(s_q, X_m)) \end{cases}$$

where s_m is the state of child $x_j \in C_u(p)$.

Finally, the downward probability $D_j(s_l)$ is the probability that all labels of a tree except for those of subtree $t(j)$ are generated and that the state of node x_j is s_l. The downward probability parameter is defined as follows.

$$D_j(s_l) = \begin{cases} \text{If } j \text{ is the root then } \pi[s_l], \\ \text{else if } j = x_{\rightarrow}(p) \text{ then} \\ \quad D_p(M_q) b[M_q, o_p] F_j(M_q, s_l) + \\ \quad D_p(I_q) b[I_q, o_p] F_j(I_q, s_l) + \\ \quad D_p(X_q) F_j(X_q, s_l). \\ \text{o.w.} \\ \quad D_p(M_q) b[M_q, o_p] F_j(M_q, s_l) \\ \quad \{a[\{M_q, s_l\}, M_m] B_k(M_q, M_m) + \\ \quad\ a[\{M_q, s_l\}, I_l] B_k(M_q, I_l) + \\ \quad\ a[\{M_q, s_l\}, X_m] B_k(M_q, X_m)\} + \\ \quad D_p(I_l) b[I_l, o_p] F_j(I_l, \pm s_l) \\ \quad \{a[\{I_l, s_l\}, M_m] B_k(I_l, M_m) + \\ \quad\ a[\{I_q, s_l\}, \pm I_l] B_k(I_q, I_l) + \\ \quad\ a[\{I_q, s_l\}, X_m] B_k(I_q, X_m)\} + \\ \quad D_p(X_q) F_j(X_q, s_l) \\ \quad \{a[\{X_q, s_l\}, M_m] B_k(X_q, M_m) + \\ \quad\ a[\{X_q, s_l\}, I_l] B_k(X_q, I_l) + \\ \quad\ a[\{X_q, s_l\}, X_m] B_k(X_q, X_m)\}. \end{cases}$$

where x_k is the younger brother of x_j and s_m is the younger brother state of s_l.

As in PSTMM, the profilePSTMM probability parameters can be calculated in a backward-breadth-first fashion from leaves to root for upward, forward and backward, and then the downward probability parameter can be calculated from the root back down to the leaves. Thus a similar Expectation-Maximization (EM) algorithm [6] to calculate the maximum likelihood is used. The pseudocode for parameter estimation is given in Figure 5.

Each parameter is calculated not only through the given tree structure but also via the structure of the state model. The pseudocode is simplified and does not specify the details for self-loop transition parameter calculations, but the basic idea is that for insertion states, the state position in the state model does not change. Note that compared to the algorithm for PSTMM, in our new algorithm, we do not need to traverse all states to call the **find** F, **find** B, **find** U, or **find** D functions since the state to evaluate is given in the arguments. The fixed state positions allow us to specify the states according to position directly. From these changes, it should be apparent that the computation time is drastically decreased.

3.3 Likelihood estimation

The likelihood for a given tree can then be calculated from a set of parameters using the Begin (which can be set as a match state) and insert states at position 0 and the upward probability for the root node:

$$L(T; \theta) = \sum_{s}^{M,I} \pi[s_0] U_1(s_0).$$

Accordingly, the likelihood of a set of trees is the product of the likelihood of each tree: $L(\mathbf{T}; \theta) = \prod_u L(T_u; \theta)$.

3.4 EM algorithm

The expectation values for π, a, and b are then computed, with which the original values can be updated using the EM algorithm [6]. We illustrate how these expectation values are calculated with one example for $\gamma(s_q, s_m, s_l)$, which is the expectation value that the state of a node is s_l and that the states of its parent and immediately elder sibling are s_q and s_m, respectively. Let us define $H_j(s_q, s_m, s_l) = F_j(s_q, s_m) U_j(s_m) a[\{s_q, s_m\}, s_l] B_k(s_q, s_l)$. Then for each state type, the calculations are as follows:

$$\gamma(\{s_q, s_m\}, M_l) = \frac{\sum_{p:C(p)} D_p(s_q) b[s_q, o_p] \sum_{j \in C(p) \setminus j_{\_}(p)} H_j(s_q, s_m, M_l)}{L(T; \theta)},$$

where x_k is the younger brother of x_j.

```
procedure calculate()
  calculate(root,beginState);
  calculateD(root,beginState);
procedure calculate(node x, state y)
  /* for all children of x and */
  /* all corresponding state children of y, oldest to youngest */
  for each c ∈ C(x) and d ∈ C(y) do
    calculate(c, d)
  /* from oldest child to youngest child */
  calculateU(eldestNode, eldestState);
  calculateFB(eldestNode, eldestState);
procedure calculateU(node x, state y)
  find U_x(y);
  /* go to immediately younger sibling */
  if x has younger sibling and y has younger state do
    calculateU(youngerNode, youngerState);
procedure calculateFB(node x, state y)
  find F_x(parent(y), y);
  if x has younger sibling and y has younger state do
    /* go to immediately younger sibling */
    calculateFB(youngerNode, youngerState);
  else /* go to immediately elder sibling */
    calculateBF(elderNode, elderState);
procedure calculateBF(node x, state y)
  find B_x(parent(y), y);
  /* go to immediately elder sibling */
  if x has elder sibling and y has elder state do
    calculateBF(elderNode, elderState);
procedure calculateD(node x, state y)
  find D_x(y);
  /* for all children c of x and all children d of y*/
  for each c ∈ C(x) and d ∈ C(y)do
    calculateD(c,d)
```

Fig. 5. Pseudocode for calculating F, B, U and D.

Similarly, for the insertion state type:

$$\gamma(\{s_q, s_m\}, I_m) = \frac{\sum_{p:C(p)} D_p(s_q) b[s_q, o_p] \sum_{j \in C(p) \backslash \mathrm{j}_{\leftarrow}(p)} H_j(s_q, s_m, I_m)}{L(T;\theta)}$$

and for deletion:

$$\gamma(\{s_q, s_m\}, D_l) = \frac{\sum_{p:C(p)} D_p(s_q) \sum_{j \in C(p) \backslash \mathrm{j}_{\leftarrow}(p)} H_j(s_q, s_m, \pm D_l)}{L(T;\theta)}.$$

In the maximization step, we update $\hat{a}$ as follows:

$$\hat{a}[\{s_q, s_m\}, s_l] = \frac{\sum_u \gamma_u(\{s_q, s_m\}, s_l)}{\sum_u \sum_{l'} \gamma_u(\{s_q, s_m\}, s_{l'})}.$$

The procedure for computing the expectation values also traverses the state model, so the computation time does not need to iterate through all combinations of states as before.

4 RESULTS

We tested our new model on both synthetically generated data and real glycan data from the KEGG GLYCAN database. Profiles are retrieved by reading the label output probabilities for all labels in the alphabet at each match position.

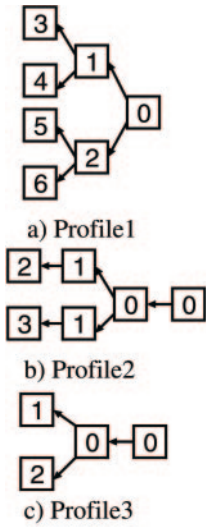


Fig. 6. Synthetic data profiles tested.

4.1 Synthetic data

In order to clearly evaluate the performance of profilePSTMM, we generated a controlled data set of tree structures containing a specific profile. We tested this on three different profiles, each of varying complexity. We then retrieved the learned profiles to see how well they compared with the original profiles. Accuracy, precision and AUC were also calculated by comparing the log likelihood values of the positive dataset with the negative dataset, which was generated based on the parent-child label distributions of the positive dataset. Accuracy is the threshold at which the positive and negative test scores are best discriminated, and precision is the proportion of the correctly predicted examples to the number of examples predicted to be positive. AUC, or the area under the ROC (Receiver Operator Characteristic) curve [12,13], is calculated by first sorting the examples by their computed likelihoods and then by Equation 1.

$$\mathrm{AUC} = \frac{R_n - \frac{n_n \cdot (n_n+1)}{2}}{n_n \cdot n_p},$$

where $n_n(n_p)$ is the number of negative (positive) examples and R_n is the sum of the ranks of the negative examples. We note that $n_n = n_p$ in our experiments. Figure 6 illustrates the profiles we tested.

4.1.1 Synthetic Experiment Setup For each profile, 50 trees were generated by the following procedure. Take the profile as a tree and randomly generate zero to two levels between the second and third levels, labeling them randomly with symbols from the set $\sigma = \{0, 1, \ldots, s\}$ where $s = 7$ for Profile1 and $s = 5$ for Profile2 and Profile3. Additionally, random siblings are added between the leaves up to three children. Taking these 50 trees as the positive data set, we also generated 50 trees for the negative data set with which to compare performance. These trees in the negative set were generated based on the parent-child label distributions of the positive set.

We also fixed the shape of the state model, as in Figure 7 (without the Begin state). For each node at the first and second levels,

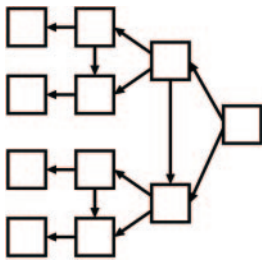


Fig. 7. State model structure for all experiments presented in this work. The Begin state has been omitted. For each node at the first and second levels, $C(_i) = 2$. For the third level nodes, $C(i) = 1$, and the leaves are of course $C_i = 0$.

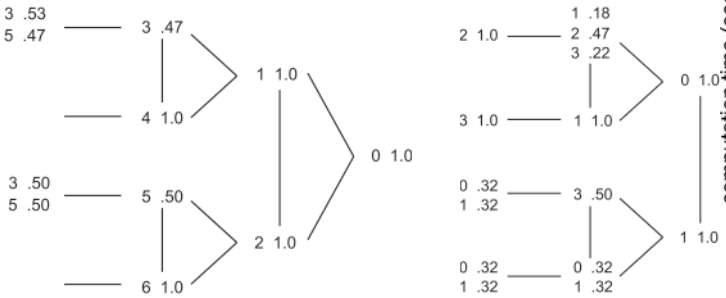


Fig. 8. Profiles learned from synthetic data. In order from left to right: Profile1, Profile2, and Profile3. Probability values below .20 were omitted.

$C(i) = 2$. For the third level nodes, $C(i) = 1$, and the leaves are of course $C(i) = 0$. This would be sufficient to account for the extra levels in the positive dataset.

4.1.2 Resulting Synthetic Profiles Learned As a result, the profiles that were learned from these three data sets are given in Figure 8. It is evident from these profiles that the eldest child most strongly learns the data and probably controls the amount of data learned. For example, the profile learned from Profile1 emphasizes 3 and 5 at the eldest leaves of both main branches. Similarly, the profile of Profile2 is learned in the elder main branch, as the younger main branch is basically random. The same can be said for Profile3.

Finally, the accuracy, precision (at sensitivity of 0.3), and AUC values of these data sets are given in Table 2. The reason that Profile2 has the worst performance may be due to the two $1 \leftarrow 0$ linkages that appear in the original profile. This causes the negative dataset to contain this linkage more frequently, thus decreasing the discrimination performance.

4.1.3 Computation Time In order to assess the efficiency of our new model, we compared the computation time of profilePSTMM with different state model sizes against PSTMM. This was performed on a Linux machine with 16GB of memory and dual processor AMD Opteron™ 250. The plot of the computation time compared with PSTMM using the same number of states is given in Figure 9. ProfilePSTMM scales much better because of the fixed structure of the state model, while PSTMM does not because of the need to traverse all pairs of states.

Table 2. Accuracy, precision and AUC values for synthetic data and N-Glycan subtype experiments. P1, P2, and P3 represent Profile1, Profile2 and Profile3, respectively

	P1	P2	P3	High-mannose	Hybrid	Complex
Accuracy	.914	.788	.892	.978	.982	.970
Precision	.843	.974	.926	.882	.904	.882
AUC	.910	.868	.903	.959	.966	.954

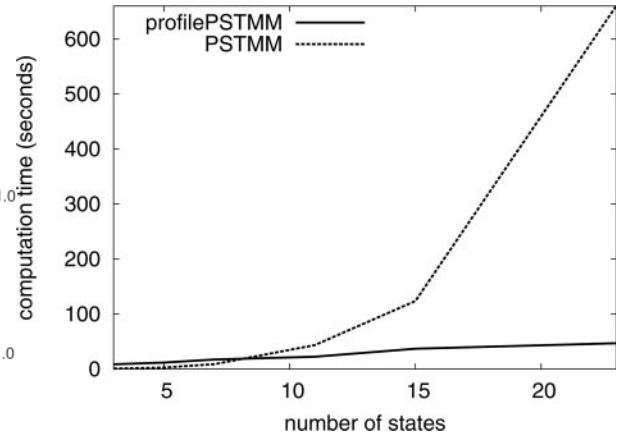


Fig. 9. Plot of computation times comparing profilePSTMM with PSTMM based on number of states. ProfilePSTMM scales much better because of the fixed structure of the state model, while PSTMM needs to traverse all combinations of states.

4.2 Glycan data

The glycan data set originated from KEGG GLYCAN, taken on February 2, 2006. Because of the variety of monosaccharide names and variations possible, a translation table was created to map variations of basic monosaccharides to the basic name for simplicity. The over 200 different names were mapped to the eight highlighted in Figure 1. Those that did not correspond well with any of these basic monosaccharides were labeled as ''Other.''

4.2.1 Initial output label probabilities In our training methodology, we initialized the label output probabilities not to random values but to those that are most often found. That is, we counted the labels appearing at the first and second levels as one set, and the labels appearing at the leaves as a second set. Based on these label distributions, we initialized the output label probabilities of our state model at the first and second levels with the first set and the leaves with the second set (with slight variations to add variability). This technique allows the model to learn from the data more easily.

4.2.2 N-Glycan subclass profiles We first manually extracted the N-Glycans from KEGG (note the word of caution in Section 2.2.2 regarding glycan classifications) and further took those that could be classified as one of the three basic subtypes: high-mannose, hybrid, and complex type N-Glycans. Figure 10 illustrates the differences between these structures. High-mannose type (left) is dominated by

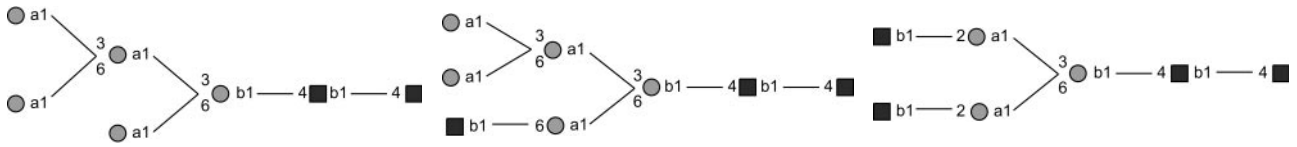


Fig. 10. The basic N-Glycan subtypes are differentiated by these structures. High-mannose type (left) is dominated by mannoses at the leaves. Complex type (right) is a combination of GlcpNAc and Galp at the leaves. Hybrid type (center) branches off with mannose on one branch and GlcpNAc and Galp structures on the other.

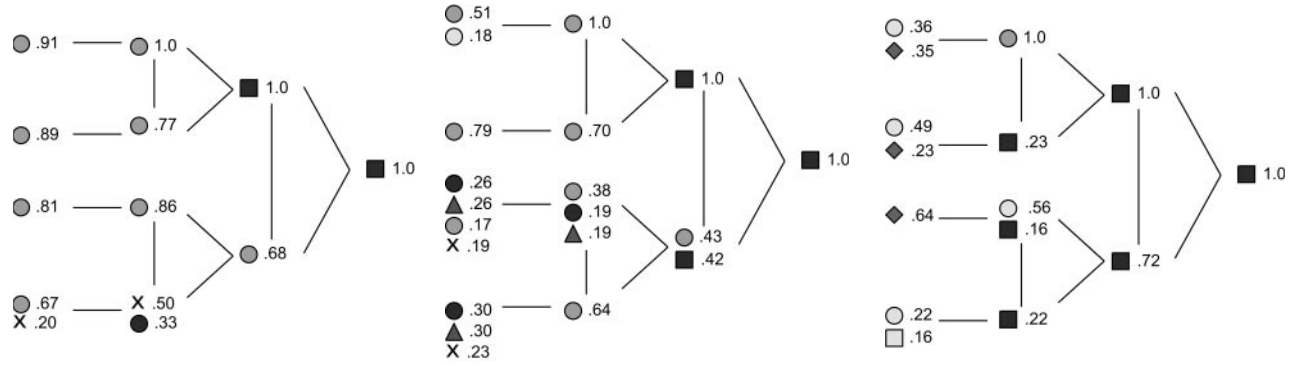


Fig. 11. Trained state models for N-Glycan subtype data. Label output probabilities <.15 are omitted. In order from left to right: high-mannose, hybrid, and complex. Just as for the synthetic data experiment, the eldest child has the tendency to learn the data and correspondingly captures the profiles of each of these subtypes.

mannoses at the leaves. Complex type (right) is a combination of GlcpNAc and Galp at the leaves. Hybrid type (center) branches off with mannose on one branch and GlcpNAc and Galp structures on the other. Doing a search using KCaM [4] with these basic structures resulted in 64 high-mannose structures, 16 hybrid structures and 351 complex structures (after manual curation for those that contained only the full N-Glycan core structure, and nothing else such as amino acids). For each subtype, we generated a dataset of 50 positive and 50 negative structures. The positive set was generated by randomly selecting a tree 50 times from the glycan subset, and the negative set would be generated using the parent-child distribution of monosaccharides from the complete positive set. This was repeated five times to set up the five-fold cross validation test. Note that in this experiment we neglected to account for the binding conformations to reduce the size of the variables, but these can be added by modifying the labels accordingly.

The performance for each of the subtypes is given in Table 2, from which it is clear that the performance is comparable to PSTMM with AUC scores in the mid-90% range. We can also easily retrieve the profiles learned, as shown in Figure 11, for each dataset by extracting the label output probabilities from the match states at each position. The legend for these structures is given in Figure 1. The label ''X'' refers to ''Other'' monosaccharides. Also note that output labels having low probabilities are omitted for clarity.

The profiles obtained again indicate the strength of the eldest child state capturing the data most confidently. The high-mannose profile does indeed capture the mannoses at the leaves, and the core GlcpNAc pair is found at the root end. The ''Other'' monosaccharides also accumulate in the lower branch. For the hybrid profile, the mannoses are well-captured by the upper branch, and the extra GlcpNAc in the bottom branch of the core corresponds well with the hybrid-type characteristic of the GlcpNAc after the core mannose. The rest of the subtree after this GlcpNAc reflects the variety of sub-structures that are found in this subclass. For the complex profile, the root end seems to have captured the GlcpNAcs that are in both the core as well as in the leaves that alternate with Gals. Indeed, we see the Gals appearing at the leaves, in addition to sialic acids (NeupAc) which are usually only found at the leaves.

4.2.3 Lectin binding glycans The purpose of this work was to analyze the glycan binding affinity of lectins. In particular, it was preferable to find sialic-acid binding affinity data. However, although sialic-acid binding lectin arrays for glycans have been developed and used for experiments [5,21] on glycan binding affinity, we found that the glycans spotted on these arrays were basically trimers, which would not be interesting enough for our purposes. Therefore, we used the data for glycan binding affinities of galectins that was published in a review by Hirabayashi *et al*. [16]. Galectins are carbohydrate-binding proteins that bind to galactose (Galp) residues. We then took those galectins that bound to larger and more varied glycans with higher affinity: galectin-3 and galectin-9N. We weighted the data set according to binding affinity by proportionately adding more of the glycans that had higher affinity. The binding affinities and corresponding weights of glycans for these two types of galectins are given in Table 3. These affinities are the normalized and inverted values from the original disassociation constants so that higher values indicate higher affinity. 30 trees were then randomly selected from the distribution of glycans in this data set. Negative data sets of the same size were also generated based on the parent-child label distribution of the trees in the positive set.

Table 3. Binding affinities and weights for Galectin-3 and Galectin-9N. Affinity values are normalized and inverted from the original data by Hirabayashi [16] such that higher values indicate higher affinity. Abbreviations: NA3: triantennary N-Glycan; fuc. NA3: core-fucosylated NA3; NA4: tetraantennary N-Glycan; fuc. NA4: core-fucosylated NA4; penta.: pentasaccharide; A-hexa: A-hexasaccharide; LN3: LAcNAc; LN5: $(LacNAc)_5$

	Gal-3 affinity (weight)	Gal-9N affinity (weight)
NA3	1.28205 (1)	2.6316 (2)
fuc. NA3	1.21951 (1)	2.2222 (2)
NA3 type1	1.08696 (1)	1.6949 (0)
NA4	1.44928 (1)	5.5556 (5)
fuc. NA4	1.40845 (1)	4.3478 (4)
Galili penta.	1.47059 (1)	0.2273 (0)
Forssman penta.	0.16129 (0)	11.111 (11)
A-hexa	1.5873 (1)	3.8462 (3)
LN3	2.85714 (2)	1.2346 (0)
LN5	5.26316 (5)	8.3333 (8)

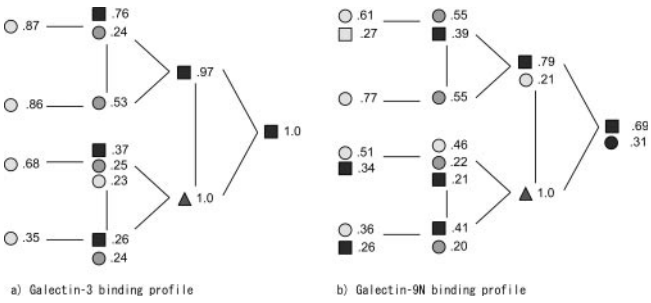


Fig. 12. Lectin binding glycan profiles. Label output probabilities <.20 are omitted. It was not surprising that the galectins appeared strongly at the leaves as the nature of galectins is to bind to galectins. We could also confirm that the Galp-GlcpNAc linkage appeared in several of the branches at the leaves, confirming the results in the literature.

The resulting profiles are given in Figure 12. It was not surprising that Galp appeared strongly at the leaves as the nature of galectins is to bind to Galp. We could also confirm that the Galp-GlcpNAc linkage appeared in several of the branches at the leaves, confirming the results in the literature. We can also explain that the Galp-Manp linkage is due to the core structure of the N-Glycans in the data set because of the GlcpNAc-GlcpNAc linkages at the root. Furthermore, it is noted that the Fucp appearing near the root with 100% probability accounts for the fucosylated core structures of the N-Glycans, and that it usually does not have children. When looking at the state transitions, indeed we find that the transitions out of this state have higher delete transitions compared with the rest of the trained state model (data not shown). Ignoring the descendants of this state, we find that our profiles capture both the N-Glycan core structures as well as the highly recognized Galp-GlcpNAc linkages at the leaves.

The summary of the accuracy, precision and AUC values for these two models are also presented in Table 4, where we can see that the discrimination of galectin-binding glycans against the negative data set is very high. Thus, we claim that there are indeed patterns that are sibling-dependent in the data which can be captured by our model. Plus we can see the profiles directly from the model.

Table 4. Performance of lectin binding glycans for Galectin-3 and Galectin-9N

	Gal-3	Gal-9N
Acc	.847	.91
Prec	1.0	.918
AUC	.93	.931

4.2.4 Glycan class differentiation As our final test of profile PSTMM, we tested the ability of our model to distinguish between different classes of glycans. Our results up to now indicated the strong influence of N-Glycan core structures appearing in the profiles. Thus we took different classes of glycans to compare their profiles with one another. In consideration of space constraints, we present the comparison between O-Glycans and sphingolipids here.

Figure 13 is a plot of the log likelihood values for glycans in the O-Glycan and Sphingolipid classes. The number of glycans in each class is given in Table 1. A model was trained for each class of glycans, and the models were tested on both classes. The dotted line represents $y = x$, to differentiate the line between the two classes. We find that the majority of glycans can be classified into the right class, except for a few in the center. Examining the cluster of glycans in the center, we found that these are glycans that can actually be classified into both classes.

The contingency matrix for these clusters as divided by the diagonal is given in Table 5. The total number of glycans that could be distinguished accurately were 923 out of the total 968 structures, resulting in a 95.4% rate of discrimination accuracy. Other pairs of classes were also tested, and similar results were obtained (data not shown).

5 DISCUSSION

We have developed a new model that performs more efficiently and conveniently for finding patterns in sibling-dependent tree structures. We integrated new types of state transitions to take into consideration the differences between the parent dependencies of children and the elder sibling dependencies of younger siblings. The fixed positions of the states also reduced the computational complexity by a factor of $O(|S|)$. We also found that the computation times decreased accordingly with fewer iterations of the EM algorithm.

To better improve the performance when training this model, we set the initial label output probability parameters to those that would be most likely found at specific positions. In particular, we set those at the root and second levels to the same distribution of labels as found in the root and second levels of the training set, and we set the leaves to the distribution of the leaves of the training set. The distributions were varied slightly at each position for variability. This procedure can be improved even more by initializing the state model to a structure that better suits the data at hand. In fact, we had configured our state model differently to test various sizes that were both more and less complex than the one presented here. We obtained similar results, capturing various

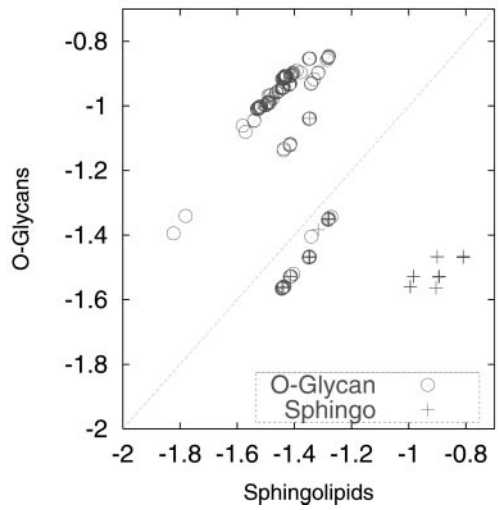


Fig. 13. Plot of log likelihood values of O-Glycan and Sphingolipid glycans using model trained on O-Glycans vs. model trained on Sphingolipids.

Table 5. 2×2 Contingency matrix for O-Glycan vs. Sphingolipid (Sphingo) class differentiation. The resulting discrimination rate is $923/968 = 95.4\%$.

	O-Glycan model	Sphingo model
O-Glycans	445	44
Sphingo	1	478

smaller patterns with smaller state models, and capturing extraneous profiles with models that were more complex. So depending on the input data, the state model should be configured appropriately. That is, the trees can first be multiply aligned by performing pairwise alignments such as with KCaM [4] and obtaining a 'generalized tree' for the aligned trees. These trees can be hierarchically aligned with one another to obtain an overall tree structure which can be used to specify the structure of the state model. This process may be more cumbersome compared to the free states in the PSTMM model, but improvement in performance is gained as a tradeoff. The label output probabilities can also be initialized based on the distribution of labels reflected in the alignment.

In our results, the profile for the galectin binding glycans captured the most common linkage that were found to be indicators of higher affinity in the literature. Although we did not consider the linkage conformations (for simplicity and for reducing the variables in the training set), these results imply that with more data, the same results can be obtained. Future work should focus on analyzing sialic-acid binding proteins once sufficiently large glycan structures and binding affinity data are accumulated. Our new model allows for a quick interpretation of such abundant data very efficiently.

Finally, it is important that future work not only focus on structural data, but also annotation and interaction data such as with proteins. As more microarray data for glycan-related enzymes such as glycosyltransferases accumulate, it should be possible to analyze the biosynthetic and degradation processes of glycans using probabilistic techniques. ProfilePSTMM is just one step towards the future of glycome informatics.

6 CONCLUDING REMARKS

Our new profile PSTMM model is a significant remodeling of the PSTMM model. The novel idea to incorporate different types of state transitions gave this model the final touch it needed. With our new algorithms for parameter estimation, not only did we decrease the computational expense of the original model, but profile extraction is now extremely straightforward. The performance of the original model is still maintained, such that long-range sibling dependencies that exist in the data can be found accurately. The trained models could also distinguish between different classes. Thus as the field of glycome informatics continues to grow and resources continue to develop, our model will surely become an important tool in analyzing these complex structures.

ACKNOWLEDGEMENTS

This work is supported in part by the Kyoto University 21st Century COE Program ''Knowledge Information Infrastructure for Genome Science'' with support from MEXT (Ministry of Education, Sports, Science and Technology), Japan.

REFERENCES

[1] K.F. Aoki, H. Mamitsuka, T. Akutsu, and M. Kanehisa. A score matrix to reveal the hidden links in glycans. *Bioinformatics*, 21(8):1457–1463, 2005.
[2] K.F. Aoki, N. Ueda, A. Yamaguchi, M. Kanehisa, T. Akutsu, and H. Mamitsuka. Application of a new probabilistic model for recognizing complex patterns in glycans. In *Proc. 12th ISMB*, 2004.
[3] K.F. Aoki, A. Yamaguchi, Y. Okuno, T. Akutsu, N. Ueda, M. Kanehisa, and H. Mamitsuka. Efficient tree-matching methods for accurate carbohydrate database queries. *Genome Informatics*, 14:134–143, 2003.
[4] K.F. Aoki, A. Yamaguchi, N. Ueda, T. Akutsu, H. Mamitsuka, S. Goto, and M. Kanehisa. KCaM (KEGG Carbohydrate Matcher): a software tool for analyzing the structures of carbohydrate sugar chains. *Nucl. Acids Res.*, 32:W267–W272, 2004.
[5] B.S. Bochner, R.A. Alvarez, P. Mehta, N.V. Bovin, O. Blixt, J.R. White, and R.L. Schnaar. Glycan array screening reveals a candidate ligand for siglec-8. *J. Biol. Chem.*, 280(6):4307–12, 2005.
[6] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.
[7] M. Diligenti, P. Frasconi, and M. Gori. Hidden tree Markov models for document image classification. *IEEE Trans. on PAMI*, 25(4):519–523, 2003.
[8] S. Doubet, K. Bock, D. Smith, A. Darvill, and P. Albersheim. The complex carbohydrate structure database. *Trends Biochem. Sci.*, 14:475–477, 1989.
[9] S.R. Eddy. Hidden Markov models. *Current Opinion in Structural Biology*, 6:361–365, 1996.
[10] S.R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998.
[11] D. Goldberg, M. Sutton-Smith, J. Paulson, and A. Dell. Automatic annotation of matrix-assisted laser desorption/ionization n-glycan spectra. *Proteomics*, 5(4):865–75, 2005.
[12] D.J. Hand and R.J. Till. A simple generalisation of the area under the ROC curve for multiple classification problems. *Machine Learning*, 45:171–186, 2001.
[13] J.A. Hanley and B.J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.
[14] K. Hashimoto, S. Goto, S. Kawano, K.F. Aoki-Kinoshita, N. Ueda, M. Hamajima, T. Kawasaki, and M. Kanehisa. KEGG as a glycome informatics resource. *Glycobiology*, in press, 2005.
[15] K. Hashimoto, S. Kawano, S. Goto, K.F. Aoki-Kinoshita, M. Kawasima, and M. Kanehisa. A global representation of the carbohydrate structures: a tool for the analysis of glycan. *Genome Informatics*, 16(1):214–222, 2005.
[16] J. Hirabayashi, T. Hashidate, Y. Arata, N. Nishi, T. Nakamura, M. Hirashima, T. Urashima, T. Oka, M. Futai, W.E. Muller, F. Yagi, and K. Kasai. Oligosaccharide specificity of galectins: a search by frontal affinity chromatography. *Biochim Biophys Acta*, 1572(2–3):232–54, 2002.

[17] J. Holgersson and J. Lofling. Glycosyltransferases involved in type 1 chain and Lewis antigen biosynthesis exhibit glycan and core chain specificity. *Glycobiology*, in press, 2006.

[18] M. Kanehisa, S. Goto, M. Hattori, K.F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucl. Acids Res.*, 34:D354–D357, 2006.

[19] Y. Koyama, T. Suzuki, S. Odani, S. Nakamura, J. Kominami, J. Hirabayashi, and M. Isemura. Carbohydrate specificity of lectins from Boletopsis leucomelas and Aralia cordate. *Biosci. Biotechnol. Biochem.*, 70(2):542–5, 2006.

[20] A. Loss, P. Bunsmann, A. Bohne, A. Loss, E. Schwarzer, E. Lang, and C. W. von der Lieth. SWEET-DB: an attempt to create annotated data collections for carbohydrates. *Nucl. Acids Res.*, 30(1):405–408, 2002.

[21] J. Stevens, O. Blixt, L. Glaser, J.K. Taubenberger, P. Palese, J.C. Paulson, and I.A. Wilson. Glycan microarray analysis of the hemagglutinins from modern and pandemic influenza viruses reveals different receptor specificities. *Journal of Molecular Biology*, 355(5):1143–55, 2006.

[22] N. Ueda, K.F. Aoki-Kinoshita, A. Yamaguchi, T. Akutsu, and H. Mamitsuka. A probabilistic model for mining labeled ordered trees: capturing patterns in carbohydrate sugar chains. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1051–1064, 2005.

[23] A. Varki, R. Cummings, J. Esko, H. Freeze, G. Hart, and J. Marth, editors. *Essentials of Glycobiology*. Cold Spring Harbor Laboratory Press, New York, 1999.

[24] A. Varki. Sialic acids as ligands in recognition phenomena. *FASEB J.*, 11: 248–255, 1997.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e35–e39
doi:10.1093/bioinformatics/btl218

The iRMSD: a local measure of sequence alignment accuracy using structural information

Fabrice Armougom[1], Sébastien Moretti[1], Vladimir Keduas[1] and Cedric Notredame[1,*]

[1]Laboratoire Information Génomique et Structurale, CNRS UPR2589, Institute for Structural Biology and Microbiology (IBSM), Parc Scientifique de Luminy, case 934, 163 Avenue de Luminy, FR-13288, Marseille cedex 09

ABSTRACT

Motivation: We introduce the iRMSD, a new type of RMSD, independent from any structure superposition and suitable for evaluating sequence alignments of proteins with known structures.

Results: We demonstrate that the iRMSD is equivalent to the standard RMSD although much simpler to compute and we also show that it is suitable for comparing sequence alignments and benchmarking multiple sequence alignment methods. We tested the iRMSD score on 6 established multiple sequence alignment packages and found the results to be consistent with those obtained using an established reference alignment collection like Prefab.

Availability: The iRMSD is part of the T-Coffee package and is distributed as an open source freeware (http://www.tcoffee.org/).

Contact: cedric.notredame@europe.com; cedric.notredame@igs.cnrs-mrs.fr

1 INTRODUCTION

The computation of accurate sequence alignments constitutes a pre-requisite for an ever increasing number of biological analyses. These include phylogenetic reconstruction, structure prediction, domain based analysis, function prediction and comparative genomics. In all these cases, the purpose of the alignment is to exploit evolutionary variations in order to reveal biologically meaningful patterns. The discovery and the proper analysis of these patterns depend entirely on the alignment correctness.

In many cases, an alignment is considered to be biologically correct when it accurately reflects the structural relationship between the considered sequences. This result is achieved by matching structurally equivalent residues. Assembling such an alignment is trivial when the sequences are highly similar but becomes harder for remote homologues. When considering alignments of sequences with less than 25% identity (the so-called twilight zone), standard scoring schemes like substitution matrices become uninformative and it can be difficult to determine the alignment accuracy, or even whether the sequences are truly related or not. So far, the most satisfying way of aligning remote homologues has been to use structural information whenever possible (Huang and Bystroff, 2006; Lesk and Chothia, 1980).

The use of structural information, however, carries its own peril, and while the sequence analysis community tends to consider structure based alignments as unambiguous and unquestionable gold standards, a closer look reveals a much less clear cut situation. More than 20 structure alignment packages have been developed (Goldsmith-Fischman and Honig, 2003). All these packages tend to produce different alignments because of their different underlying optimization algorithms. Furthermore, the lack of a universally accepted criterion for describing the quality of a structural alignment makes it difficult to determine the relative merits of all these packages (Kolodny, *et al*., 2005). The most common procedure to evaluate structure superpositions is to use the root mean square distance deviation (RMSD) of superposed atoms. This measure estimates the mean square distance between the equivalent alpha carbons of the two superposed structures. It can be ambiguous because of its dependence on two critical parameters: the minimization method and the procedure used to exclude structurally non equivalent regions (loops for instance).

Having several methods that deliver structure based sequence alignments and not knowing which one does best is a major issue in a context where structure-based alignments are routinely used to improve and guide the development of sequence alignment methods (Wallace, *et al*., 2005). A direct consequence of this situation has been the development of at least five collections of reference structure based sequence alignments (Edgar, 2004; Mizuguchi, *et al*., 1998; O'Sullivan, *et al*., 2004; Raghava, *et al*., 2003; Thompson, *et al*., 2005; Van Walle, *et al*., 2005). These collections are all used for a similar purpose: the benchmark of sequence alignment algorithms. Since it is virtually impossible to compare these datasets and decide whether some are more informative than others, the most common practice is to use them all, and look for common trends in the global results (Katoh, *et al*., 2005).

While results measured on these reference collections tend to agree for datasets with more than 30% identity, variations appear when considering sets of remote homologues (Katoh, *et al*., 2005). Aside from potential accuracy problems, the simplest explanation for these discrepancies is the possibility for alternative sequence alignments to be structurally equivalent, especially when considering remote homologues (Lackner, *et al*., 2000). In this context, setting one specific alignment as a reference becomes an arbitrary choice and therefore a bias toward specific alignment methods. In practice, the authors try to minimize that effect by specifying the core regions that should be used for the comparison, but this choice is also difficult and somehow arbitrary. We suggest in this paper that replacing the reference alignments with an RMSD measure would

*To whom correspondence should be addressed.

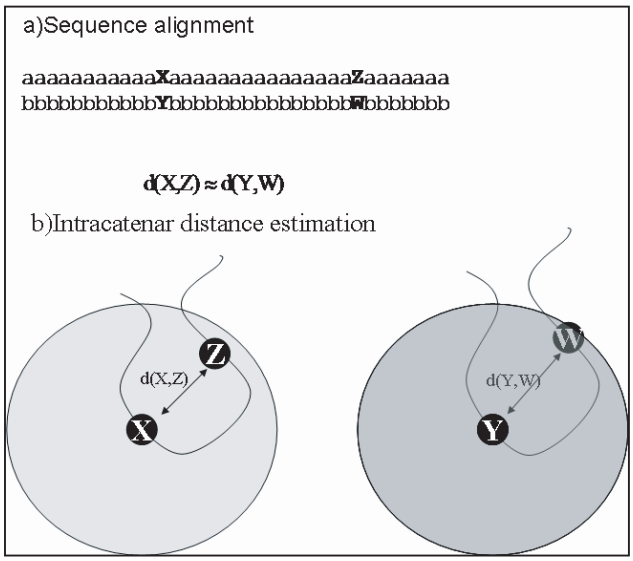


Fig 1. Basic principle of the iRMSD. Equivalences implied by the sequence alignment are tested on the structure. The assumption is that if XY and ZW are correctly aligned, then the distance between residues XZ and YW must be similar. ZW pairs are only considered if they are within a sphere of radius *R*, centered on X and Y.

be a more objective way to evaluate the sequence alignments of proteins. The RMSD has two advantages over standard methods: no dependence on a reference alignment and the possibility to quantify the structural correctness of any protein sequence alignments (provided the protein structures are known). The main drawback, however, is the reliance of the RMSD on a structure superposition strategy. This key step affords many alternative solutions whose relative merits are difficult to estimate (Kolodny, *et al*., 2005).

We redesigned the RMSD measure to make it independent from any structure superposition procedure. We named this measure iRMSD because it is an RMSD based on intra-molecular distance comparisons. The iRMSD is a follow up of the APDB measure (O'Sullivan, *et al*., 2003), designed to evaluate alignments for their compatibility with the structural superposition they imply. While APDB was a complex measure depending on three semi arbitrary parameters, the new iRMSD algorithm only requires one parameter. We show here that the iRMSD behaves just like a standard RMSD both numerically (values range) and structurally (similar structural meaning). We finally show that a straightforward normalization makes the iRMSD perfectly suitable for evaluating and comparing sequence alignment methods without the need of pre-established reference alignment collections.

2 METHODS

2.1 The iRMD measure

The iRMSD measure follows the underlying principle of APDB: given a correct alignment of two protein sequences A and B (Figure 1), if X is aligned with Y and Z with W, then the XZ distance (d(XZ)) must be similar to d(YW). The better the alignment of A and B, the smaller the average difference between all possible pairs d(XZ) and d(YW). The iRMSD associated with the aligned pair X and Y is estimated by considering every aligned pair Z and W within a sphere of radius (r) centered on X and Y that verifies the equation:

$$\mathrm{d(XW)} < r \ \textbf{AND} \ \mathrm{d(YZ)} < r \tag{1}$$

The ensemble of pairs ZW that verify equation 1 is named the neighborhood and noted N(XY). The default value of r is 10 Å (O'Sullivan, *et al*., 2003), which corresponds to a neighborhood size of 20-40 residues. The local iRMSD can be estimated as follows:

$$iRMSD(XY) = \sqrt{\frac{\sum_{ZW}(d(XZ)-d(YW))^2}{N(XY)}} \tag{2}$$

The summation is made over all the aligned ZW pairs within the neighborhood (Equation 1). Pairs XY with an empty neighborhood have their local iRMSD undefined. The global measure is obtained by summing on every pair XY and dividing by the number of pairs with a non empty neighborhood (N):

$$iRMSD = \frac{\sum_{XY} iRMSd(XY)}{N} \tag{3}$$

The iRMSD thus defined is not suitable for comparing alternative alignments, as it tends to give a better score to alignments with long gaps and few well aligned residues. In order to simultaneously take into account the superposition accuracy and the extent of the alignment (i.e. the number of matched residues), we adapted the CI formula of Kleywegt and Jones (Kleywegt and Jones, 1994) to turn the iRMSD into a Normalized iRMSD (NiRMSD):

$$NiRMSd = \frac{iRMSD^{*}MIN(L1,L2)}{N} \tag{4}$$

L1 and L2 are the respective lengths of the two sequences, and N the number of residue pairs with a non empty neighborhood. This formula amounts to incorporating a gap penalty that deals with indels and aligned pairs whose neighborhood is empty.

2.2 Validation procedure using Prefab

We used the Prefab (Edgar, 2004) collection of reference alignments to analyze the iRMSD. Prefab is an extensive collection of 1682 pairwise structural alignments obtained by combining the output of two structure alignment programs: CE (Shindyalov and Bourne, 1998) and DALI (Holm and Sander, 1993). In each of these alignments the authors have defined core regions where the DALI and the CE methods agree and have used these regions for evaluation purpose. Given one Prefab reference alignment and an alternative target alignment of the same sequences, the Qscore is defined as the fraction of core columns in the reference alignment found aligned identically in the target. In order to evaluate multiple sequence alignment packages, Prefab also includes in each dataset a collection of about 48 sequences homologous to the two structures. When evaluating an MSA package, the large dataset is aligned and the Qscore is measured on the core regions of the induced alignment of the two structures.

We evaluated the RMSD and the iRMSD of Prefab alignments. However, because of various inconsistencies between the ATOM, the SEQRES fields of the PDB entries and the sequences of the Prefab alignments, LSQMAN could only handle 587 of the original Prefab entries. This sample had roughly the same identity distribution as the entire Prefab (243 dataset having with than 20% identity (on the reference Prefab alignment), 172 between 20 and 40% identity and 171 with more than 40% identity). We believe it to be representative and large enough for the purpose of the present analysis.

2.3 Evaluation of the standard RMSD

We used the LSQMAN package (Kleywegt and Jones, 1999) to estimate the standard RMSD associated with the Prefab alignments. The local RMSD was estimated by superposing the residues contained in a window of size 21 (2∗10+1) centered on a pair of aligned residues. The superposition was

made using the Xalignment function of the LSQMAN package. The overall RMSD was obtained by sliding the window and averaging over all the windows.

2.4 Multiple sequence alignment methods

We benchmarked the iRMSD measure on the alignments produced using the public distributions of six multiple sequence alignment packages: ClustalW (Version 1.83) (Thompson, *et al*., 1994), DialignII (Version 2.2.1) (Morgenstern, 1999), Muscle (Version 3.6) (Edgar, 2004), Mafft (Version 5.6) (Katoh, *et al*., 2005), ProbCons (Version 1.10) (Do, *et al*., 2005) and T-Coffee (Version 3.75) (Notredame, *et al*., 2000).

2.5 Availability

The iRMSD package is part of the t_coffee package. It is an open source freeware that can be downloaded on http://www.tcoffee.org/. It comes along with an extensive documentation.

3 RESULTS

We started by comparing the iRMSD with the standard RMSD. We did so by measuring the scores associated with the 587 Prefab alignments. The measurements were either made on core regions (Figure 2a) or on the entire Prefab Alignments (Figure 2b). Both figures indicate a very strong correlation between the two measures. The core analysis gave an r^2 correlation coefficient of 0.92 while the measure on the entire alignments gave an r^2 of 0.93. As expected, the dispersion increases with the RMSD values. The Prefab alignments are high quality structure based alignments, but we also checked the behavior of the methods when analyzing alignments of lower quality (Figure 2c). We selected the Dialign method whose alignments have an average Prefab Qscore of 0.65 on the entire dataset (0.32 in the [0-20] identity range). Figure 2c shows that the two measures remain correlated up to an RMSD of 2.5 Å ($r^2 = 0.75$), indicating a saturation of the iRMSD measure for values above 1.6 Å. This apparent saturation is a consequence of the different local substructures compared by each method (windows for the RMSD and sphere for the iRMSD) and it does not occur when measuring the standard RMSD on spheres of radius 10 Å rather than on windows. When doing so the correlation is very good ($r^2 = 0.91$ over the full range, data not shown).

We further checked the local aspect of the measures by plotting both the local iRMSD and the local RMSD against several Prefab alignments. The 1aoh_1anu example is displayed on Figure 3 and clearly shows that both measures are well coordinated all along the alignment. While the iRMSD indicates two narrow peaks not found in the RMSD, both methods agree on the final series of peaks. We used LSQMAN to superpose the two structures and were satisfied to find that the peaks showing in the iRMSD curve effectively correspond to regions poorly superposed. Although the iRMSD seems to reveal more sharply these locations, it is fair to say that the standard RMSD could probably be parameterized to yield similar results (for instance by lowering the window size).

Having established that the iRMSD behaves like a standard RMSD measure we then estimated whether that measure is suitable for evaluating the relative accuracy of multiple sequence alignment packages. For that purpose, we aligned the Prefab datasets with six MSA methods and for each of these methods we evaluated the Qscore, the Normalized iRMSD (NiRMSD, Equation 3) and estimated the fraction of alignments having a NiRMSD better or

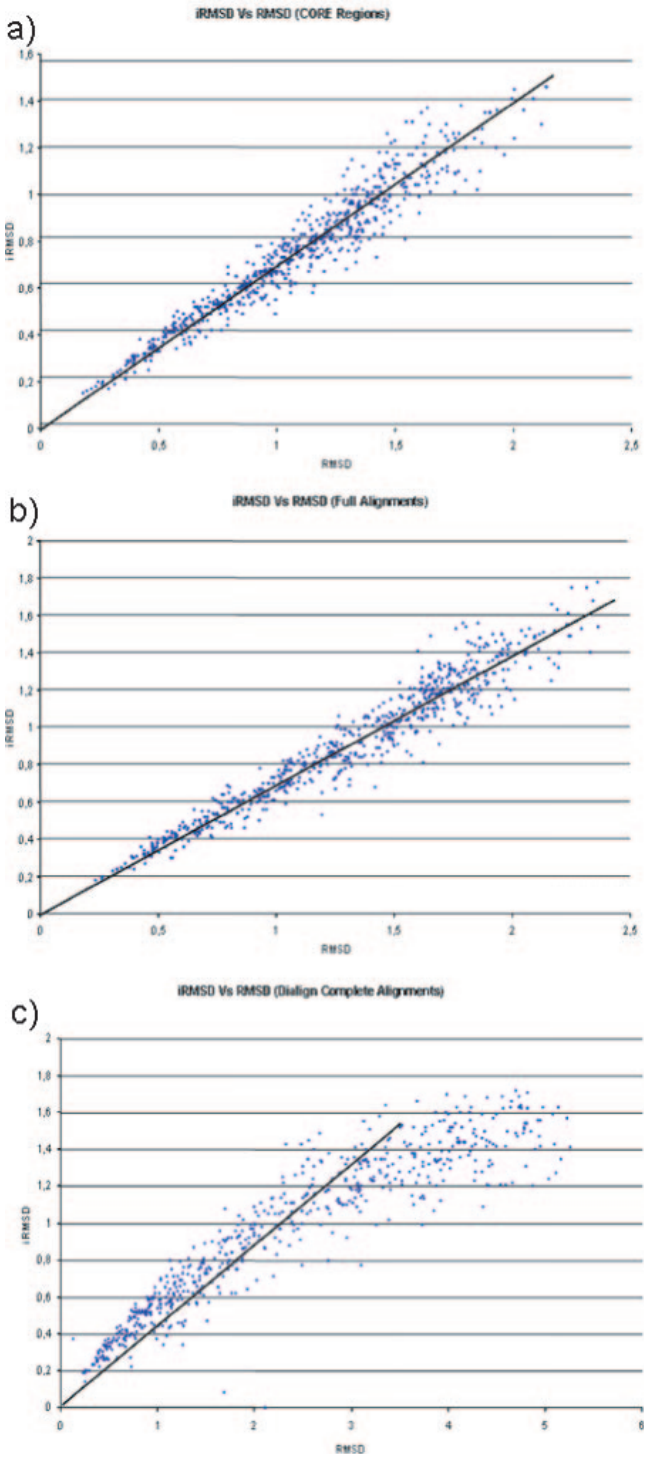


Fig 2. Correlation between the iRMSD and a standard LSQMAN RMSD. 1a) RMSD versus iRMSD of 587 of Prefab reference Alignments. The (i)RMSDs were only measured on the regions annotated as core in Prefab. The iRMSD is on the vertical axis and the regular RMSD, as obtained from LSQMAN, is on the horizontal axis. Each dot corresponds to one dataset. 2a) RMSD versus iRMSD on 587 Prefab reference Alignments. The (i)RMSDs were measured on the entire alignments. 2c) RMSD versus iRMSD on 587 Prefab datasets, aligned by Dialign. The dataset is the same as before and the (i)RMSDs were measured on the entire alignments.

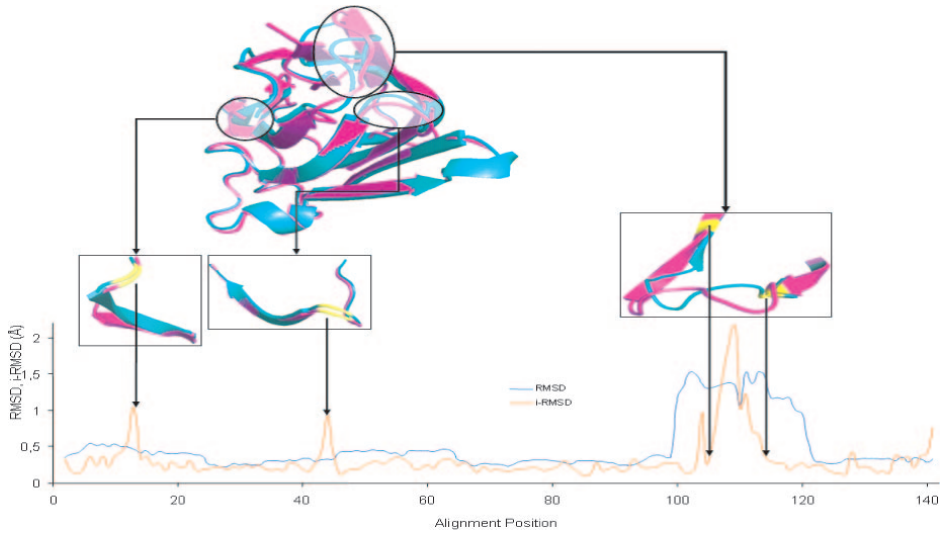


Fig 3. Local Comparison of the iRMSD against a *standard LSQMAN RMSD*. The comparison was made on the Prefab reference alignment of 1aohA_1anu. The two structures were superposed by LSQMAN (1aohA: violet, 1anu:blue). The alignment was then evaluated locally using either LSQMAN to measure the RMSD (Blue line) or T-Coffee/iRMSD to measure the local iRMSD. The (i)RMSDs values were plotted on the vertical axis against the alignment positions. Portion of the superposition corresponding to the peak were extracted and encapsulated.

Table 1. Average Qscore

Range	N	Dialign	Clustal	Muscle	TCoffee	ProbC.	MAFFT	PREFAB
0-20	243	0.32	0.34	0.43	0.44	0.48	**0.49**	----
20-40	171	0.80	0.83	0.86	0.87	**0.88**	**0.88**	----
40-100	173	0.96	0.96	0.97	**0.98**	0.97	**0.98**	----
Total	587	0.65	0.67	0.71	0.73	0.74	**0.75**	----

a) **Average Qscore**: Range is the range of identity of the considered Prefab datasets, as measured on the reference alignments. N is the number of Prefab datasets in each range. Dialign, ClustalW, Muscle, TCoffee, ProbCons and Mafft are the average Qscores as measured on the alignments produced by these packages. The entries corresponding to the best performance for each category are underlined and in bold. The best Qscore are the highest.

Range	N	Dialign	Clustal	Muscle	TCoffee	ProbC.	MAFFT	PREFAB
0-20	243	3.46	2.10	1.82	2.16	1.85	**1.76**	0.85
20-40	171	0.91	0.82	0.80	0.79	**0.77**	**0.77**	0.67
40-100	173	0.44	0.58	0.44	0.44	0.44	**0.43**	0.43
Total	587	1.83	1.28	1.11	1.25	1.12	**1.08**	0.67

b) **Average NiRMSD**: The labels are the same. The measure is the average NiRMSD as measured on the core regions of the alignments. The Prefab column corresponds to the evaluation of the Prefab reference alignments. The best NiRMSD scores are the lowest.

Range	N	Dialign	Clustal	Muscle	TCoffee	ProbC.	MAFFT	PREFAB
0-20	243	0.02	0.10	0.05	0.09	0.06	**0.10**	----
20-40	171	0.36	0.36	0.46	0.56	**0.57**	**0.54**	----
40-100	173	0.86	0.89	0.89	**0.92**	0.89	0.91	----
Total	587	0.36	0.40	0.42	0.47	0.45	**0.47**	----

c) **Best NirRMSD** Fraction: fraction of alignments having a NiRMSD better or equal to the Prefab reference as measured on the core regions. The labels are the same.

Table 2. Consistency between the NiRMSD and the Qscore

Range	Npair	Consistent	Inconsistent
0-20	7290	0.86*	0.14*
20-40	5130	0.90*	0.10*
40-100	5190	0.94*	0.06*
Total	17610	0.90*	0.10*

a) Core Regions: Range is the range of identity of the considered Prefab datasets, as measured on the reference alignments, Np is the number of pairs on which the comparison was carried out. Consistent is the fraction of pairs for which the Qscore and the NiRMSD score were consistent. For the purpose of this table, two pairs were considered consistent whenever their Qscore differed by less than 1 point percent and their NiRMSD by less than 0.05 Å. A binomial test was carried out on the results and entries marked with * indicate results whose p-value is lower than 0.000001.

Range	Npair	Consistent	Inconsistent
0-20	7290	0.79*	0.21*
20-40	5130	0.84*	0.16*
40-100	5190	0.84*	0.16*
Total	17610	0.82*	0.18*

b) Same as a) but with the NiRMSDs measured on the entire alignments.

equal to the Prefab reference (Best NiRMD fraction), as measured on the core regions.

The results (Table 1a,b and c) are unambiguous and clearly show a high correlation between the Qscore, the average NiRMSD and the Best NiRMSD fraction. As expected, the Prefab reference alignments outperform every other method (Table1b, Prefab), with a NiRMSD always lower than the rest, especially in the distant homologue category (Table 1b, Prefab, [0-20]). The rankings suggested by each score are in broad agreement when considering equivalent lines in each table. We looked at the statistical signifi-

cance of all these analyses. For doing so we considered every dataset individually and estimated the consistency between the Qscore and the NiRMSD measured on two alternative alignments. For instance, given a dataset and two alignments (aln1 and aln2) generated by two different methods, the Qscore and the NiRMSD are consistent if they indicate the same relationship between the two alignments (*e.g.* aln1 better than aln2 according to Qscore AND NiRMSD).

This measure was used to analyze every possible pair of methods (Table 2a,b). The results show that Qscore and NiRMSD are highly correlated with 90% consistency between the two measures on core regions and 82% when considering entire alignments. The correlation is not affected by the level of identity between the considered sequences. These figures were measured on more than 17000 pairs of alignments. We checked these results for statistical significance, using a binomial test and assuming an equal probability of 0.5 for consistency and inconsistency. The results are highly significant on each category, with P-Values systematically lower than 10^{-6}. These results confirm that the NiRMSD measure is at least as discriminative as Prefab.

CONCLUSION

We describe the iRMSD, a measure with all the advantages and properties of a standard RMSD without requiring any structure superposition. A simple normalization makes it possible to use the iRMSD for evaluating the accuracy of structure based sequence alignments. This measure, named NiRMSD, was applied on the alignments produced by 6 popular multiple sequence alignment packages. In 90 % of the cases the NiRMSD measure was in agreement with the Prefab ranking (Qscore). These findings, highly significant from a statistical point of view, suggest the suitability of this new measure for evaluating sequence alignments accuracy whenever structural information is available. We also expect that the method can easily be extended to sequences having a close homologue with a known structure.

Future developments will involve applying the iRMSD to Multiple Structure Alignment analysis. We are also planning to use the NiRMSD measure to compare structure alignment packages and check whether some methods clearly outperform the others or whether some structure alignment meta-method should be designed instead. Further refinement could also involve exploring the capacity of the iRMSD measure to automatically identify and exclude unalignable positions.

ACKNOWLEDGEMENTS

The development of this project was supported by CNRS (Centre National de la Recherche Scientifique), Sanofi-Aventis Pharma SA., Marseille-Nice Génopole and the French National Genomic Network (RNG). We thank Prof. Jean-Michel Claverie (head of IGS) for useful discussions and material support. We also thank Dr Phillip Bucher who provided many of the original ideas through useful discussions.

REFERENCES

Do,C.B., Mahabhashyam,M.S., Brudno,M. and Batzoglou,S. (2005) ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res*, **15**, 330–340.

Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, **32**, 1792–1797. Print 2004.

Goldsmith-Fischman,S. and Honig,B. (2003) Structural genomics: computational methods for structure analysis. *Protein Sci*, **12**, 1813–1821.

Holm,L. and Sander,C. (1993) Protein structure comparison by alignment of distance matrices. *J Mol Biol*, **233**, 123–138.

Huang,Y.M. and Bystroff,C. (2006) Improved pairwise alignments of proteins in the Twilight Zone using local structure predictions. *Bioinformatics*, **22**, 413–422.

Katoh,K., Kuma,K., Toh,H. and Miyata,T. (2005) MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Res*, **33**, 511–518.

Kleywegt,G.J. and Jones,T.A. (1994) Superposition. *CCP4/ESF-EACBM Newsletter Protein Crystallog*., **31**, 9–14.

Kleywegt,G.J. and Jones,T.A. (1999) Software for handling macromolecular envelopes. *Acta Crystallogr D Biol Crystallogr*, **55** (**Pt 4**), 941–944.

Kolodny,R., Koehl,P. and Levitt,M. (2005) Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J Mol Biol*, **346**, 1173–1188.

Lackner,P., Koppensteiner,W.A., Sippl,M.J. and Domingues,F.S. (2000) ProSup: a refined tool for protein structure alignment. *Protein Eng*, **13**, 745–752.

Lesk,A.M. and Chothia,C. (1980) How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. *J Mol Biol*, **136**, 225–270.

Mizuguchi,K., Deane,C.M., Blundell,T.L. and Overington,J.P. (1998) HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci*, **7**, 2469–2471.

Morgenstern,B. (1999) DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment [In Process Citation]. *Bioinformatics*, **15**, 211–218.

Notredame,C., Higgins,D.G. and Heringa,J. (2000) T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, **302**, 205–217.

O'Sullivan,O., Suhre,K., Abergel,C., Higgins,D.G. and Notredame,C. (2004) 3DCoffee: combining protein sequences and structures within multiple sequence alignments. *J Mol Biol*, **340**, 385–395.

O'Sullivan,O., Zehnder,M., Higgins,D., Bucher,P., Grosdidier,A. and Notredame,C. (2003) APDB: a novel measure for benchmarking sequence alignment methods without reference alignments. *Bioinformatics*, **19 Suppl 1**, i215–221.

Raghava,G.P., Searle,S.M., Audley,P.C., Barber,J.D. and Barton,G.J. (2003) OXBench: A benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics*, **4**, 47.

Shindyalov,I.N. and Bourne,P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, **11**, 739–747.

Thompson,J., Higgins,D. and Gibson,T. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **22**, 4673–4690.

Thompson,J.D., Koehl,P., Ripp,R. and Poch,O. (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins*, **61**, 127–136.

Van Walle,I., Lasters,I. and Wyns,L. (2005) SABmark--a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, **21**, 1267–1268.

Wallace,I.M., Blackshields,G. and Higgins,D.G. (2005) Multiple sequence alignments. *Curr Opin Struct Biol*, **15**, 261–266.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e40–e48
doi:10.1093/bioinformatics/btl225

A model-based approach for mining membrane protein crystallization trials

Sitaram Asur[1], Pichai Raman[2], Matthew Eric Otey[1] and Srinivasan Parthasarathy[1,*]

[1]Department of Computer Science and Engineering, Ohio State University and [2]Department of Biophysics, Ohio State University

ABSTRACT

Motivation: Membrane proteins are known to play crucial roles in various cellular functions. Information about their function can be derived from their structure, but knowledge of these proteins is limited, as their structures are difficult to obtain. Crystallization has proved to be an essential step in the determination of macromolecular structure. Unfortunately, the bottleneck is that the crystallization process is quite complex and extremely sensitive to experimental conditions, the selection of which is largely a matter of trial and error. Even under the best conditions, it can take a large amount of time, from weeks to years, to obtain diffraction-quality crystals. Other issues include the time and cost involved in taking multiple trials and the presence of very few positive samples in a wide and largely undetermined parameter space. Therefore, any help in directing scientists' attention to the hot spots in the conceptual crystallization space would lead to increased efficiency in crystallization trials.

Results: This work is an application case study on mining membrane protein crystallization trials to predict novel conditions that have a high likelihood of leading to crystallization. We use suitable supervised learning algorithms to model the data-space and predict a novel set of crystallization conditions. Our preliminary wet laboratory results are very encouraging and we believe this work shows great promise. We conclude with a view of the crystallization space that is based on our results, which should prove useful for future studies in this area.

Contact: Srinivasan Parthasarathy, 693 Dreese Lab, 2015 Neil Ave, Columbus, OH-43210, USA, Email: srini@cse.ohio-state.edu

1 INTRODUCTION

The study of membrane proteins is one of prime importance in all branches of proteomics. Membrane proteins are integral to all cellular functions acting as mediators between the cell and its environment. These remarkable proteins play important roles in energy transduction, cell signaling, and maintaining the integrity of the cells' internal environment. However, there is still very little known about their function since many of their structures remain unknown. Since structure leads to function, discovering the structure of these proteins will help lead to understanding their function and will aid in creating drugs for a host of diseases. However, compared to soluble proteins, there is a dearth of membrane proteins with known structure. In order to obtain the structure of a protein with high resolution, many scientists rely on the powerful technique of X-ray diffraction, which requires a protein crystal. However, obtaining good quality crystals of membrane proteins is an arduous task when compared to water soluble proteins. This is due to the fact that membrane proteins typically get trapped as an intractable aggregate during the crystallization process, limiting access to their structure (Caffrey, 2003).

The science of crystallization is still quite preliminary and there is very limited knowledge on what actually causes crystallization to occur. Hence, crystallographers are forced to systematically sift through a wide parameter space (for example, physio-chemical, biophysical, biological parameters) to grow crystals with good diffraction characteristics. This trial-and-error approach (not unlike searching for needles in a haystack) has been shown to be difficult due to the phenomenally large cost and time requirements to perform the crystallization experiments.

As a consequence, the set of conditions currently employed is based almost entirely on earlier experimental successes (Rupp, 2003). These conditions, while not random, are not specifically designed for a particular protein. From a statistical perspective, this amounts to over-sampling certain regions in the multi-dimensional crystallization space. Such screens represent what is known as a sparse matrix. These sparse matrices assume that different proteins will crystallize under the same conditions. This assumption is not completely valid (Rupp, 2003). Therefore, researchers have attempted to vary one or two of the chemical components from successful combinations to obtain new favorable conditions. Unfortunately, this has met with mixed success, requiring many trials to get a few good crystals.

The process of protein crystallization involves using a protein/lipid membrane that is mechanically mixed and brought to correct water content and temperature. At this stage, suitable chemical reagents are added and protein crystals are then allowed to form. The reagents can be grouped into classes such as precipitant, additive, buffer, and detergent. The temperature, type and concentration of the lipid and reagents are of utmost importance in protein crystallization. These physio-chemical conditions and reagents together form the crystallization screen. Additionally, the current hypothesis is that the optimum conditions (those that cause the best resolution crystals) are protein-specific. Overall, it is fairly difficult to obtain crystals of any quality. With this in mind, it stands to reason that if we can produce a greater number of conditions that do in fact bring

*To whom correspondence should be adderessed.

about crystals, we can assume some percentage of them will have crystals of good diffraction quality. Obtaining multiple crystals is also good since this adds to robustness and reliability of the results. Specifically, the end goal is to develop a screen (with different crystallization conditions) that is optimal for a particular protein and maximizes the number of high-resolution crystals.

In this case study, we consider the crystallization space to be broadly classified into three areas, mapped to classes 0, 1 and 2. These are analogous to the three levels, clear, precipitate and crystalline, proposed by Kimber *et al*. (2003). The 'hot spots' are the areas that yield protein crystals (class 2). A large part of the space consists of clear or 'no-hit' areas that are not conducive for the production of crystals (class 0). There are also areas that do not yield crystals but produce protein precipitates (class 1).

Some researchers (Rupp, 2003; Segelke, 2001) discuss the virtues of random sampling on the crystallization space. We believe a more structured and intelligently designed approach will lead to success. In this work, we examine the use of suitable supervised learning algorithms to examine relationships or correlations between the input parameters (protein properties, crystallization conditions) and model the response output (crystals, precipitates or no crystals) for existing trials and then close the loop to identify interesting 'hot spots' (areas with high potential for yielding good quality crystals) in the space for future trials. We use the model learnt to predict the outcomes for a randomly sampled set of conditions. We then perform stratified sampling based on our model, incorporating physio-chemical constraints, to obtain new sets of conditions to test in the wet laboratory. Our premise is that this method is more structured and a more profitable option than random sampling. Preliminary wet lab experiments seem to validate this premise. Our results also allow us to hypothesize a view of the crystallization space. We provide details of this hypothesis at the end of the paper. To summarize, the main contributions of this paper are:

- Application of supervised learning algorithms to model the protein crystallization space.
- Model-based prediction and stratified sampling to obtain novel conditions with high probability of yielding crystals.
- A hypothetical view of the crystallization space based on our results.

2 BACKGROUND ON PROTEIN CRYSTALLIZATION

In this section, we provide some background on the crystallization process and discuss some related work in this area.

2.1 Cubic phase (*In meso*) crystallization

Crystallization is essentially a phase separation technique in a thermodynamically stable system, with the favorable outcome being the formation of a crystal. There are a host of techniques currently employed to crystallize proteins. The basis of this project rests on the laurels of a relatively new technique for membrane protein crystallization known as the Cubic Phase or *in meso* method (Caffrey, 2003). This is the technique from which all our data is derived. The cubic phase technique is based on the assumption that the protein to be crystallized is initially reconstituted into the lipid bilayer of the cubic phase (Caffrey, 2003). The essential steps involved in this technique are adding a protein/lipid membrane that is mechanically mixed and brought to correct water content and temperature to form the cubic phase. At this stage, additives and precipitants are added and protein crystals can then form in a time span that extends from hours to months. This can be done manually or with the aid of a robot for high throughput crystallization. While the technique itself seems straightforward, the way in which the *in meso* method crystallizes proteins is still not well understood. There is a great deal of speculation as to how this method works. The crux of the method rests on the understanding of the peculiar phase behavior of lipids. Lipids have two standard phases, liquid and solid. However, they also possess a third set of phases known as liquid-crystalline phases. These phases represent configurations of the lipid molecules in aqueous medium that arise due to the amphipathic nature of lipids and the hydrophobic effect. A set of lipid phases is shown in Figure 1.

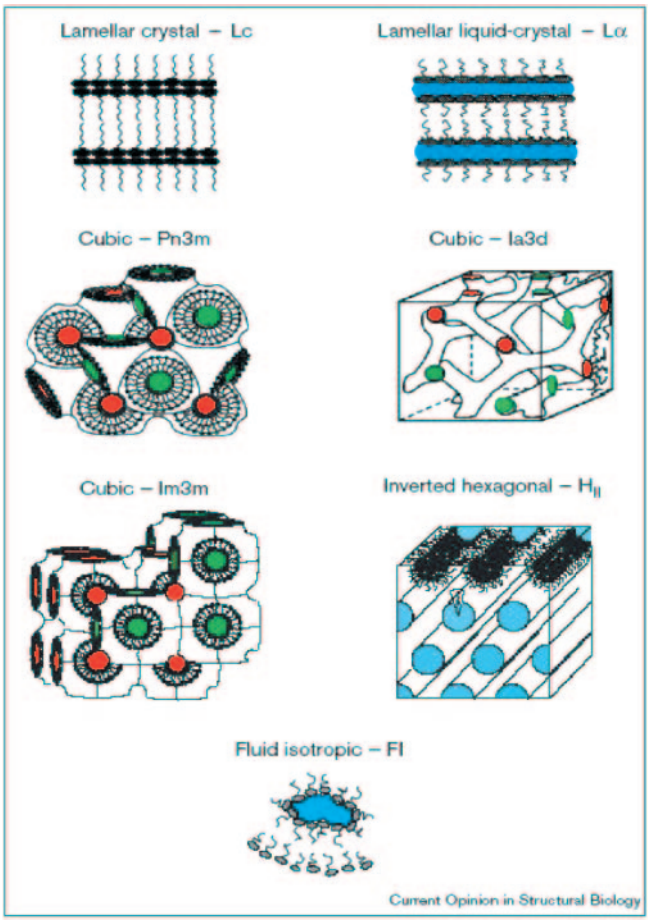


Fig. 1. Lipid Phases—Cubic (Pn3m, Ia3d, Im3m), lamellar liquid-crystal (L), and inverted hexagonal (HII) phases are the liquid crystalline phases. Fluid isotropic (FI) is a liquid phase and lamellar crystal (Lc) is a solid phase (Caffrey, 2003).

The lipid phases change with the water content and temperature and this is plotted out in a Temperature/Composition (T/C) diagram. An understanding of these phases is of utmost importance since in this method one must achieve the cubic phase, hence utilizing the right proportion of water (to the protein/lipid blend) and the right

temperature is key in order to get to the appropriate phase. The idea is that during the mixing process, the proteins start off solubilized in detergent micelles but then reconstruct into the lipid bilayer with the introduction of dry lipid. The lipidic phase they are thrust into is the cubic phase. These phases (Pn3m, Ia3d, and lm3m) are shown in Figure 1. With the addition of salt the curvature of this phase increases, which in turn causes the protein to leave and associate in a transient lamellar phase, also shown in the figure. The belief is that as proteins leave the lamellar phase, they arrange in a highly ordered fashion and form crystals.

There exist a large number of variables in this technique such as additive structure, additive concentration, detergent type, protein structure, etc. It is not known which of these parameters are instrumental in obtaining a favorable outcome, realizing a crystal. Furthermore, researchers have varying, inconsistent and largely incomplete information as to why proteins crystallize in the first place. As a consequence, the set of conditions scientists currently employ is based almost entirely on earlier successes and the chemicals readily available currently. These conditions, while not random, are not specifically designed for a particular protein. Therefore, the likelihood of getting crystals from these screens for novel proteins is incredibly small.

2.2 Related work

The work by Samudzi *et al*. (1994) postulates that the response surface was composed of a set of disjoint clusters, rather than a single coherent cluster. Subsequently, they apply a clustering algorithm on the Biological Macromolecule Crystallization Database, which is a large collection of successful crystallization trial conditions. Their initial attempt revealed interesting qualitative relationships between recorded parameters but did not yield how this information could be used in the design of future experiments. A limitation of these experiments (along with others at the time), is that the data used consisted of only successful trials.

Several researchers (Jurisica *et al*., 2001; Kimber *et al*., 2003; Rupp, 2003), argue convincingly that a comprehensive information repository for crystal growth experiments (both positive and negative trials) is fundamental to the computational analysis of trials. This stored information is necessary to discover general rules or principles underlying the growth process for crystals, as well as to guide the reasoning algorithm for planning experiments. As noted by the above researchers, the application of data mining and knowledge discovery algorithms to such datasets is still in its infancy. Carter and Carter (1979) were one of the first to propose the use of statistical sampling techniques for this problem. Segelke (2001) assesses crystallization screens in terms of sampling and shows the advantages of random sampling. We believe that random sampling may not be the best solution since it does not use the available prior knowledge effectively. Currently, most approaches taken by crystallographers rely on either random or stratified sampling of the crystallization space. Rupp (2003) also argues that in a high throughput environment, with a large number of data points and limited prior knowledge, a semi-automated machine learning/data mining driven approach is absolutely essential. In spite of these works discussing the use of data mining algorithms, to the best of our knowledge there has been no prior work in this direction.

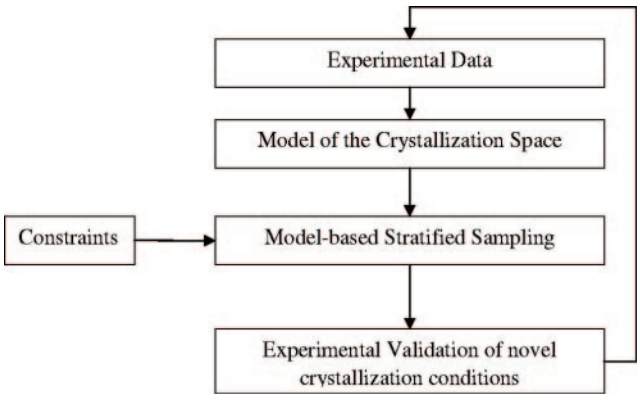


Fig. 2. Mining the Protein Crystallization space.

3 OVERVIEW OF OUR APPROACH

The protein crystallization space has been conceptualized as a high-dimensional hypercube (Rupp, 2003) with axes represented by the chemical components and other parameters. The various crystallization condition trials are obtained by sampling this space. Our strategy for mining the protein crystallization space is a closed loop consisting of four stages, represented in the flowchart in Figure 2.

- **Experimental Data:** The data obtained from prior experiments is used as training data. This data consists of sets of conditions that have been employed before in crystallization trials. An issue with using prior data is that it has been obtained almost completely from the same regions in the crystallization space. These regions have been over-sampled repeatedly. Another issue is the large bias present in the dataset, with a significant majority of the samples resulting in failures. We discuss the characteristics of the dataset in detail in the next section.
- **Modeling the space:** The empirical training data is used to build supervised models on the protein crystallization space. We believe that supervised learning algorithms such as classifiers are useful for this problem as they can use the training data and known class values to partition the space efficiently. Hence, we apply traditional classifiers and build an ensemble using the best classifiers to increase the precision of prediction. It is important to note that this approach will initially be limited since the empirical data currently available represents only a few regions. A large amount of the space is presently unknown. However, our strategy is dynamic and incremental. As we iterate, more regions of the space will be added into our training data for modeling. We present details of our modeling technique in Section 5.
- **Model-based Stratified sampling with constraints of the condition space:** We use the model of the data-space to lead us to the right regions for sampling, and the classifiers that we trained to predict class values of novel conditions. We perform stratified sampling on the predicted conditions to overcome the over-sampling issue. The objective is to discover new regions in the space that have not been visited earlier and that have high potential for yielding crystals. Our approach is iterative and incremental. At each iteration, we broaden our search space. We use stratified sampling on our results in order to maintain balance. We also need to minimize the number of

Table 1. Sample chemical conditions

0.1 M Tris HCl pH 8.5,15% iso-Propanol,0.2 M Ammonium Acetate
0.1 M Cacodylate pH 6.5,20% (wv) PEG-1000,0.2 M MgCl2
0.1 M Hepes pH 7.5,22% wv Polyacrylic Acid 5100,0.02 M Mg Chloride
0.1 M Tris Hydrochloride pH 8.5,25% wv PEG 3350,0.2 M Mg Chloride

conditions to be tested and ensure a high success-rate (reduce false positives). We leverage this by using a relatively strict metric for prediction.

At the same time, we need to consider constraints (physio-chemical, physical and biological) of the crystallization space. These constraints may be a factor of the conditions or internal parameters such as temperature and solubility. We present details of our sampling scheme in Section 6.

- **Experimental Wet lab Validation of novel conditions:** Once novel conditions have been discovered, we need to test them experimentally. One of the issues with experimentation is the expense, in terms of time and effort, for each crystallization trial. We validate the sampled conditions, again considering constraints of the crystallization process. The results from this step feeds back into the first step of the next loop. Our experimental validation results are presented in Section 7.

4 DATASET PROPERTIES

The initial data that was used to build the models for prediction was a set of screens of 3 proteins—vitamin B_{12} receptor (BtuB), bacteriorhodopsin (bR) and light-harvesting complex II (LH2) with a set of 3 monoacylglycerol(MAG) lipids—9.9 MAG, 7.7 MAG, and 9.7 MAG and a set of 480 standard conditions that originate from Hampton research, a company that specializes in developing products for biological macromolecular crystallization (http://www.hamptonresearch.com/). We used the Hampton kit for this work, since it has been shown to crystallize proteins in the past. Furthermore, members of the Caffrey lab have performed experiments to evaluate the compatibility of the Hampton screens with the cubic phase (Cherezov *et al*., 2001). This is better than using new kits which would require more extensive testing to evaluate their compatibility to the cubic phase.

The data corresponds to crystallization trials for 5 protein/lipid combinations. Each protein/lipid combination consists of 5 screens, each consisting of 96 conditions and their corresponding scores. There are 99 conditions overall with no scores, which we ignored. Hence, the data we considered finally consisted of 2301 trial conditions (5 protein/lipid combinations $\times$ 5 screens $\times$ 96 conditions each—99 elements where no data taken) with various protein, lipid, buffer, additive, precipitant combinations. Some sample conditions are illustrated in Table 1. Each protein/lipid mix was put through these conditions on a set of five 96 well plates. Each plate was then manually scored with a number from 0-9, indicating the phase/protein condition. This designation is referred to as the crystal rating.

The chemical conditions include a main buffer, a precipitant and one or more additives. The purpose of the buffer component in a screen is to cover a certain pH range (and thus charge distribution) on the protein, independent of the other components and the pH of the original protein solution'. Buffers with different pH values can thus be considered different. There are two major types of precipitants, high molecular weight poly-alcohols (like PEGs) and salts. The additives used may be buffers, precipitants or any chemical that might help crystallization. Each sample in the dataset contains a crystal rating. The crystal ratings are formulated as follows, 0-2 means lamellar or dispersed phase, 3-5 indicates protein precipitate, and 6-9 indicates the formation of crystals. The number of 0's in the dataset are very high and as the rating increases, the number of samples having that value decreases. The number of samples rated 9 is very low. To perform adequate classification, we require a better distribution. Hence, values between 0 and 2 are assigned to class 0, between 3 and 5 are assigned to class 1, and values between 6 and 9 are assigned to class 2. Class 2 is the desired class indicating the formation of crystals.

Table 2. Crystal class percentages in the dataset

Crystal classes	Number of samples	Percentage
0	1995	86.7
1	170	7.3
2	136	6

The percentages of the three discretized ratings in the dataset are given in Table 2. It can be seen that data is significantly biased with around 87% of the samples classified as 0. In this work, we treat the data samples as normal categorical data. This is a safe assumption since in this application, two buffers with different pH values can be expected to behave differently. Each sample is a feature vector of size 6 consisting of:

- Protein—Btub, bR or LH2
- Lipid—9.9 MAG, 7.7 MAG or 9.7 MAG
- Buffer—Eg. 0.1 M Na Acetate pH 4.6
- Main Precipitant—Eg. 0.5 M Magnesium Formate
- Additive—Eg. 2 M Na Chloride
- Class Value—0, 1 or 2

5 MODELING THE PROTEIN CRYSTALLIZATION SPACE

As we mentioned earlier, the protein crystallization space can been represented as an n-dimensional hypercube with axes represented by the chemical components and other parameters. The regions in this space that yield crystals are called 'hot spots'. For a given protein, there exist a large number of conditions which do not lead to precipitates. In some conditions, proteins precipitate but do not form crystals. We use supervised learning (classification) to model the protein crystallization space using the empirical data. We believe that classification is a good method to partition the data space and predict class values for new samples.

5.1 Supervised learning algorithms

In this section, we present details of the supervised learning algorithms we use.

Naive Bayes Classifiers. Naive Bayes classifiers are based on Bayes' rule of conditional probability. It uses all attributes and allows them to make contributions to the decision as if they were all equally important and independent of one another. The classifier can be formally defined as

$$C(F) = argmax_c p(C = c) \prod_{i=1}^{n} p(F_i = f_i \mid C = c) \qquad (1)$$

where c is the class and f_i are the features(attributes).

Decision Tables. Decision table classifiers are rule-based classifiers that are typically used not only for prediction but visualization of the attribute space (Kohavi, 1995). A decision table generally has two components, a scheme and a body. The scheme is the list of attributes that are used to predict the class variable. The body consists of a set of assigned values for each attribute in the scheme. The class variables that are of the same type fall into a broad category called cells. The dataset is sorted by the broadest possible field (or field with the least number of attribute types). From here, the set of rows with the same type in each attribute are grouped together in a cell. Generally, the rules of constructing a decision table involve mapping all possible combinations of the attribute space to class values. This ensures that every single input vector will have been assigned some designation. The program then simply runs through the table with the input vector to determine which class variable is appropriate.

Random Forests. A Random Forest is an ensemble classification technique which is popular due to its high accuracy. In this method, several classification trees are constructed by sampling with replacement from the original training data. In order to find the best split at a node in the tree, m random attributes are chosen and the one with the best split among them is used. Furthermore, the trees that are constructed are not pruned. Classification is done using each tree to separately classify the test data. Finally, the majority of the votes from each tree is chosen to be the prediction on the test sample.

Classification Based on Associations. This is a technique based on association rule-based classification (Liu *et al*., 1998), which can be used effectively on discrete datasets. Association rules identify collections of data attributes that are statistically related in the underlying data. An association rule is of the form $X \rightarrow Y$ where X and Y are disjoint conjunctions of attribute-value pairs. The support of the rule is the observed frequency of X and Y, $Pr(X, Y)$. The confidence of the rule is the observed frequency of Y given X, $Pr(Y \mid X)$. Given a database of transactions, a minimal confidence threshold, and a minimal support threshold, the goal of association rule mining is to find all association rules whose confidences and supports are above the corresponding thresholds. In this case, each row of the dataset can be considered to be a separate transaction, with the values in each column being the items for that transaction. The Apriori algorithm (Agrawal and Srikant, 2000) is a commonly used algorithm for mining association rules. The algorithm discovers rules for dependencies between the elements that are frequent, i.e., satisfy some minimum support and minimum confidence constraints. We then use these frequent rules to perform classification. We have tried different values of minimum support and confidence thresholds. We find that using low support (5%) and high confidence (60%) thresholds are adequate for discovering association rules even for large datasets.

Nearest Neighbor Voting. In this technique, we build separate nearest-neighbor classifiers for each attribute. For each attribute i with value v_i, we identify the k rows in the dataset that contain a value closest to the value v_i. Then we use the class values predicted by these k rows to compute a single vote value. We take the mode of the k classifications as the single vote value. This process is repeated for each attribute (v_i) resulting in several single vote values. To tally the vote values, we once again use the mode to predict the class of that sample.

Support Vector Machines. Support Vector Machines (SVMs) (Joachims, 1999; Vapnik, 1995) are based on the concept of decision planes that define decision boundaries. A decision plane separates a set of objects having different class memberships. Support Vector Machines are particularly suited to handling classification tasks that involve complex decision planes, as opposed to linear classification. They work by constructing hyperplanes in a multidimensional space. The classifier maps the input vectors to a higher dimensional space, after which it finds a linear separating hyperplane with the maximal margin in the high-dimensional space.

For our experiments, we used two popular SVM packages, SVMLight (Joachims, 1999) and BSVM (Hsu and Lin, 2002). SVMLight works efficiently for two-class problems while BSVM performs well for multi-class classification problems. We used the default linear kernel function in our experiments.

PNRule. PNRule, proposed by Joshi *et al*. (Joshi *et al*., 2001), is a rule-based classifier designed to handle skewed class distributions. PNRule works in two phases. In the P phase, it discovers positive rules that cover the target class. In the N phase, it generates rules on the negative class to eliminate false positives from the samples covered in the P phase. The rules are based on single attribute values. The test samples are run through the positive and negative rules. Accordingly, a test sample is classified positive only if it is found to satisfy a positive rule and no negative rules.

5.2 Metric

Since our goal is to discover novel trial conditions using classification, we are really interested in measuring how many of the positively predicted samples are actually positive. In other words, the precision of prediction is the key. If pos_{pred} are the samples that are predicted to be positive and pos_{actual} are the samples that are actually positive, the precision is given by

$$Precision = \frac{\| pos_{pred} \cap pos_{actual} \|}{\| pos_{pred} \|} \qquad (2)$$

We would like to point out that the accuracy of classification in this case is not particularly useful. This is due to the fact that a naive classifier that predicts class 0 for every sample will yield a high accuracy of 87% (due to the significant bias in the dataset).

5.3 Classification results

5.3.1 The bias problem: We split up the crystallization dataset randomly into training and test sets. As we mentioned earlier, the crystallization dataset consists of a large majority (87%) of negative samples. This causes significant bias and affects classifiers such as CBA and Nearest Neighbor and causes all the predictions to be of class 0.

The problem of learning with biased data has been addressed in several works in the data mining literature. As we mentioned earlier, PNRule, the rule-based classifier was proposed to handle skewed class distributions. We have implemented PNRule but find that for our dataset, the negative rules we discover cover all the samples. Hence, we cannot obtain any positive predictions. We have tried varying the negative rules based on recall, as was suggested, but do not obtain any improvement in the results.

The main methods suggested for balancing skewed training data include downsampling the non-target class, upsampling the target class and generating new samples of the target class. SMOTE (Chawla *et al*., 2002) is a technique that generates new samples of the target class using existing positive samples that are close to each other. This is possible only if there is a valid distance metric to find nearest neighbors in the set of samples, which is not true for our data. Also, in our data, the minority class is very sparse with respect to the majority class. Hence, the application of SMOTE results in a mixture of the classes (over-generalization) which is very hard to separate. Batista *et al*. (2004) evaluated different techniques for balancing training data and found that random over-sampling of the target class performs well in most cases. Using this notion, we develop an ensemble approach to eliminate the bias problem. We generate several random sub-samples of the negative class and merge each of them with over-sampled positive examples. This results in several balanced subsets of the original data. We then train our classifiers on each sub-dataset separately and use each of them to predict the class values of the test data. Finally, we use majority vote decision fusion to combine the predictions of each of the individual classifiers. We obtain much better results using this approach, although it does not completely eliminate the bias problem.

5.3.2 3-class prediction: We predict class values using all the classifiers we reviewed earlier. We perform 5-fold cross-validation. The best individual classifier is the Decision Table Classifier with a precision of 58%. The other classifiers mis-classified several samples of class 1 as class 2. The results are presented in Table 3.

5.3.3 2-class prediction: Although, we obtain a precision of 58% for the Decision Table Classifier, most of the classifiers had trouble separating the samples in the 3-class case. An interesting observation we made with the results is that a large number of samples belonging to class 1 were falsely identified as class 2. We leverage this observation as follows. In the training phase, we consider all samples of class 1 to be of class 2. Although this does not remove the bias, it increases the percentage of samples belonging to class 2 from 6% to 13%. We believe that this leads to better partitioning by the classifiers. We therefore predict once again on the test sets, using this assumption.

We find the improvement in precision to be substantial. Every individual classifier is found to predict more accurately under this scenario. The results of 5-fold cross-validation by all the classifiers are presented in the 3rd column in Table 3. CBA produced dramatic improvement (15% to 65%). The Naive Bayes technique also improved phenomenally (although its performance is still below par). The three best classifiers are, in order, Decision Table, CBA and Support Vector Machines.

5.3.4 Ensemble classification: We constructed an ensemble classifier using these three individual classifiers to improve the precision of prediction. If x_i is the test sample, and p_j where j=1..3 are the predictions from the three individual classifiers, the ensemble prediction is given by

$$Ens(x_i\pm, p_1, p_2, p_3) = \begin{cases} 2 & if\ p_1 = p_2 = p_3 = 2; \\ 0 & if\ p_1 = 0 \cup p_2 = 0 \cup p_3 = 0. \end{cases} \quad (3)$$

When we use the ensemble classifier to predict values for the test-sets we obtain a precision close to 100%. However, the number of positively predicted samples is very low (5–10). This is due to our constraint that all three individual classifiers need to predict a positive result for a sample to be classified positive. This assumption can be relaxed. Accordingly, we proceed to choose samples which any two of the classifiers predicted as positive. This gives us a larger number of positive samples (20–30) and a precision of 86% on the test data after cross validation.

Table 3. Individual Classifier Results for 3-class and 2-class cases

Algorithm	Precision (3-class)	Precision (2-class)	Percentage improvement
Naive Bayes	4%	44%	1000%
Decision Table	58%	72%	24.14%
Random Forest	35%	48%	37.14%
Bagging	52%	60%	15.38%
CBA	15%	65%	333.33%
NNV	12%	21%	75%
SVM	39.5%	65%	64.5%

6 MODEL-BASED STRATIFIED SAMPLING

As mentioned earlier, our goal in this work is not only to model the crystallization space but to discover novel regions to sample for positive conditions. Earlier works focused entirely on randomly sampling the crystallization space. Random sampling alone does not ensure success. We believe a more intelligently designed approach can yield better performance. Random sampling maximizes the variance of the data space. We propose a more principled approach, applying domain knowledge to sampling, similar to ideas proposed by Bailey-Kellogg and Ramakrishnan (2001).

We employ a two-stage stratified sampling technique in this regard. In the first step, we generate a large number of random samples. We ensure that these samples are sufficiently different from the samples in the training data. The samples are then pruned using the help of a domain expert, to enforce physio-chemical constraints such as compatibility between chemicals. We use the classifiers with the best performance to predict the class values of these samples based on the training data. The classifiers partition the samples into regions of class 0, 1 and 2. In the second step, we perform stratified sampling on the results of the classifiers. We propose two schemes that can be used for stratified sampling, depending on the context:

- In the 2-class scheme, we over-sample regions that are predicted to be class 2 or class 1 and under-sample the region that is predicted to be class 0.

- In the 3-class scheme, we over-sample regions predicted to be class 2 and under-sample the other two classes. The proportion of samples chosen that are of class 0 is less than the proportion of samples chosen that belong to class 1.

The sampled conditions are then fed to an automated robot that conducts the experiments using these conditions. During this process, constraints on internal parameters, such as temperature and solubility (Ksp values), are applied. Currently, this process is manually done by domain experts. We use two kinds of filters in the process, one to remove incompatible chemical combinations and improbable factor levels (excessive precipitant concentrations, high PEG concentrations etc) and the other to remove conditions that are not very novel.

7 EXPERIMENTAL WET LAB VALIDATION

We proceeded to generate a large set of random samples using the conditions from the Hampton kit. Since the time for experimentation is a major bottleneck in the crystallization technique, we chose to perform some preliminary experiments using a single protein/lipid combination and using the predictions of a single classifier. We used the protein Btub, which is an integral membrane protein (Chimento *et al*., 2003), and the lipid was 7.7 MAG. The rest of the conditions were randomly generated from the data, i.e., a random buffer, precipitant, and additive were chosen from the set of unique elements in the Hampton kits. Each vector was compared with the 480 Hampton kit conditions to ensure there were no duplicates.

We chose the decision table classifier, since it outperformed all the others for 3-class classification. The set of feature vectors was run through the algorithm and each sample was assigned a crystal rating. Samples were chosen for experimental validation using the 3-class scheme.

We obtained 96 conditions and conducted crystallization experiments in our laboratory using these. We used the buffers, precipitants, and salts available in the Hampton Research kit. The rest of the required reagents were prepared to specified concentrations and pH (when applicable) in house. The protein was combined with the lipid to form the cubic phase using mechanical mixing. A robot was then used to mix the reagents and dispense both the well conditions and the protein/lipid combination. All screens were set in 96 well plates which were scanned at various time intervals for crystals using a light microscope.

We found that 37 conditions, out of the 96 we tested for, produced crystals. This was close to our expectation, considering the decision table classifier yielded a precision of 58% for the 3-class problem and the presence of experimental errors. Among the hits, the crystals ranged in size from 50 microns to 90 microns (Figures 3 and 4). Interestingly, a large number of the negative trials yielded protein precipitates (class 1).

We tested our ensemble 2-class classifier on this set of experimentally determined samples. When we used the ensemble on the 96 conditions, we obtained 13 positive predictions. Since the precision of the ensemble classifier was 86%, once again considering experimental errors, we expected to get crystals in at most 8 or 9 of these trials. We were pleased to get crystals in 8 of the 13 trials. Furthermore, we were pleasantly surprised to obtain precipitates (class 1) in the negative trials. This is equivalent to a 100% precision in the 2-class scenario. Given that the number

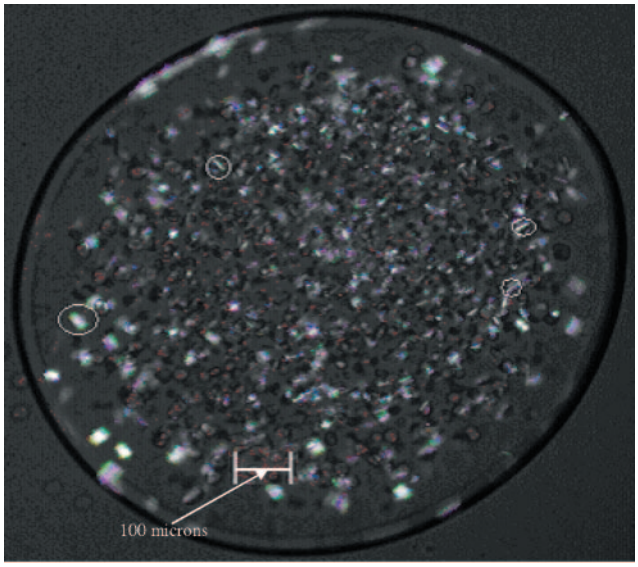


Fig. 3. BtuB crystals produced from decision table crystallization. Crystals are circled in white.

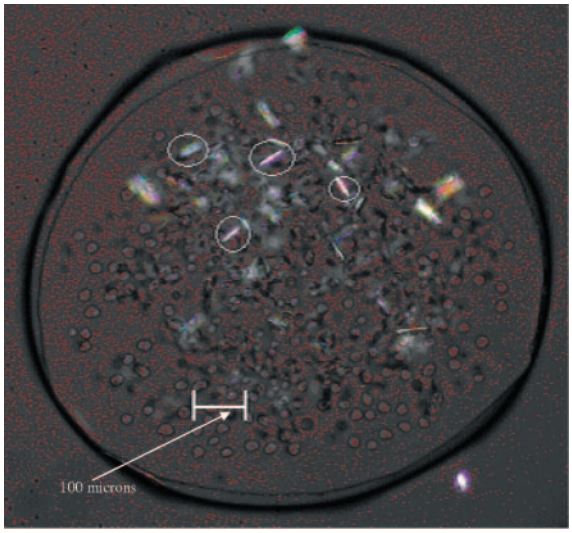


Fig. 4. More BtuB crystals produced from decision table crystallization. Crystals are circled in white.

of crystals generally obtained from crystallization screens are very few and the trials typically consume a large amount of time, our results are useful. To illustrate this, we compare the average number of positive samples from each of the 5 Hampton protein-lipid combinations with our results using the samples predicted by the Decision Table classifier. The difference can be observed in Figure 5. H1-H5 represent the 5 protein-lipid combinations in the Hampton screen kit. The average number of positive samples/96 wells for all 5 Hampton screens is around 5, whereas we obtain 37 crystals using just one screen of 96 conditions. We would also like to point out that the conditions we obtained were adequately novel when compared to the conditions in the Hampton kit which over-sampled the same space.

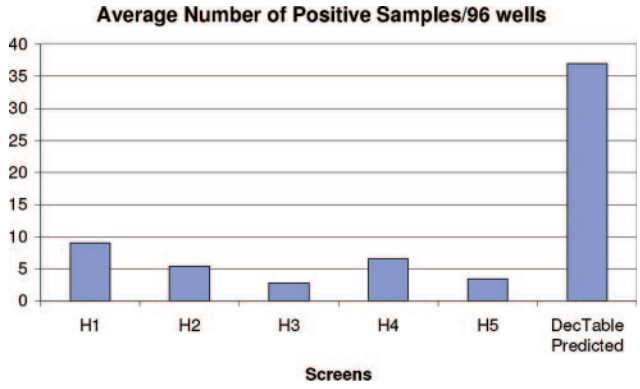


Fig. 5. Comparison between the Hampton screens and our predicted screen.

To follow-up, we plan to conduct experiments on a larger scale, generating a large set of random vectors and using our ensemble classifier to obtain predictions. We will then set up trials on these conditions in our wet laboratory. We expect to obtain favorable results as before.

7.1 Discussion

Our preliminary results are encouraging. However, it is important to note some limitations to this study. While applying data mining algorithms to build models for crystallization conditions prediction is a good approach and provides many benefits, we are limiting ourselves by using a single crystallization dataset which does not represent a truly random or evenly distributed sample of the crystallization space.

For instance, the Hampton kit contains crystallization trials information for three proteins. Using this information, we can predict conditions for only these three proteins. Since different proteins react differently to the same condition, it may not practical to make predictions for other proteins. The same holds true for the chemicals that have been used. We can create new combinations of these chemicals and predict for them. However, it is impossible to predict for sets of chemicals that have not been used in the dataset. Thus, to accurately sample it is important to develop a screen that covers the entire space. In this work, we advocate an incremental approach to this problem, with each iteration of the loop leading us slowly towards greener pastures for sampling.

A related issue is the large amount of time required for experimentation. Despite refinements over the years, this still remains the greatest bottleneck in the crystallization process. This minimizes the amount of experimental validation that can be performed. In our work, we have tried to minimize the number of conditions to be tested and decrease the false positive rate.

An important observation we have made from this study is that conventional distance metrics cannot adequately capture the distance between similar conditions in the protein crystallization space. This is supported indirectly by the poor performance of the Nearest Neighbor algorithm. This demonstrates a need for distance metrics that are sensitive to the domain, as suggested by Aggarwal (2003). A suitable distance function, in this case, would need to consider the physio-chemical characteristics of the reagents used as well as correlations between them.

The difference in precision between the 2-class classification and 3-class classification indicates that regions that yield precipitates (class 1) are close to hot spots. This was supported by the fact that the 3-class classifiers mis-classified a large number of samples belonging to class 1 as class 2 samples. Even in the wet lab experiments, we were surprised to find that a large number of samples predicted to be class 2 belonged to class 1. Our observations suggest the following view on the protein crystallization space:

- Areas fertile for crystallization (hot spots—class 2) are often well separated. This is somewhat evidenced by the poor performance of the nearest neighbor classifier.
- These areas are surrounded by areas which are not good enough to produce crystals but yield precipitates (class 1). A large part of the space comprises of no-hit areas which do not yield any crystal (class 0).

We can therefore hypothesize that the crystallization space is of a continuous nature with 2's turning into 1's and then 0's. This representation should prove to be useful for future studies.

8 CONCLUSION

In this paper, we utilize supervised learning techniques to explore the properties of the protein crystallization space and to identify potential hot spots of protein crystallization. This problem has baffled scientists for many years due to a limited understanding of the crystallization space, and the cost of performing crystallization experiments. In this work, we presented an incremental, closed-loop approach using stratified sampling and constraints to mine the crystallization space effectively for novel conditions. Our hypothesis that the crystallization space is conducive to the use of supervised learning is borne out by our classification results. Our wet lab experimental results, although preliminary, show great promise. In the future, we plan to conduct more experiments on a larger scale. We also plan to develop alternative distance metrics for the crystallization space to increase the quality of our classification techniques in hopes of refining our preliminary map of this space and finding more hot spots.

ACKNOWLEDGEMENTS

We would like to thank Dr. Martin Caffrey, Vadim Cherezov and the Caffrey lab for facilities and help provided for this work. This work is supported primarily by the DOE Early Career Principal Investigator Award No. DE-FG02-04ER25611 and also by NSF CAREER Grant IIS-0347662.

REFERENCES

Aggarwal,C. (2003) Towards systematic design of distance functions for data mining applications. *SIGKDD*, 9–18.

Agrawal,R. and Srikant,R. (1994) Fast algorithms for mining association rules. *VLDB*.

Bailey-Kellogg,C. and Ramakrishnan,N. (2001) Ambiguity directed sampling for qualitative analysis of sparse data from spatially-distributed physical systems, *IJCAI*.

Batista,G.E.A.P.A., Prati,R.C. and Monard,M.C. (2004) A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explorations Newsletter*, **6**(1), 20–29.

Caffrey,M. (2003) Membrane protein crystallization. *Journal of Structural Biology*, **142**, 108–132.

Carter Jr,C.W. and Carter,C.W. (1979) Protein crystallization using incomplete factorial experiments. *J. Biol. Chem*, **254**, 12219–12226.

Chawla,N.V., Bowyer,K.W., Hall,L.O. and Kegelmeyer,W.P. (2002) Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence and Research*, **16**, 321–357.

Cherezov,V. *et al*. (2001) Crystallization Screens: Compatibility with the lipidic cubic phase for in meso crystallization of membrane proteins. *Biophysical Journal*, **81**, 225–242.

Chimento,D.P. *et al*. (2003) Crystallization and initial X-ray diffraction of BtuB, the integral membrane cobalamin transporter of Escherichia coli. *Acta Crystallographica Section D*, **59**(3), 509–511.

Hsu,C.W. and Lin,C.J. (2002) A comparison on methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, **13**, 415–425.

Joachims,T. (1999) Making large-scale svm learning practical. *Advances in Kernel Methods—Support Vector Learning*, MIT Press.

Joshi,M.V., Agarwal,R.C. and Kumar,V. (2001) Mining needle in a haystack: classifying rare classes via two-phase rule induction. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, **30**(2), 91–102.

Jurisica,I. *et al*. (2001) Intelligent decision support for protein crystal growth. *IBM Systems Journal*, **40**(2), 394–409.

Kimber,M. *et al*. (2003) Data mining crystallization databases: knowledge-based approaches to optimizing protein crystal screens. *Proteins*, **51**, 562–568.

Kohavi,R. (1995) The power of decision tables. *Proceedings of the European Conference on Machine Learning*, 174–189.

Liu,B., Hsu,W. and Ma,Y.M. (1998) Integrating classification and association rule mining. *Knowledge Discovery and Data Mining*, 80–86.

Rupp,B. (2003) Maximum likelihood crystallization. *Journal of Structural Biology*, **142**, 162–169.

Samudzi,C.T., Fivash,M.J. and Rosenberg,J.M. (1994) Cluster analysis of Biological Macromolecular Crystallization database. *Journal of Crystal Growth*, **123**, 47–58.

Segelke,B. (2001) Efficiency analysis of sampling protocols in protein crystallization screening. *Journal of Crystal Growth*, **232**, 553–562.

Vapnik,V.N. (1995) The Nature of Statistical Learning Theory, Springer.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e49–e57
doi:10.1093/bioinformatics/btl242

Integrating structured biological data by Kernel Maximum Mean Discrepancy

Karsten M. Borgwardt[1,*], Arthur Gretton[2], Malte J. Rasch[3], Hans-Peter Kriegel[1], Bernhard Schölkopf[2] and Alex J. Smola[4]

[1]Institute for Computer Science, Ludwig-Maximilians-University Munich, Germany, [2]Max Planck Institute for Biological Cybernetics, Tübingen, Germany, [3]Graz University of Technology, Austria and [4]National ICT Australia, Canberra, Australia

ABSTRACT

Motivation: Many problems in data integration in bioinformatics can be posed as one common question: Are two sets of observations generated by the same distribution? We propose a kernel-based statistical test for this problem, based on the fact that two distributions are different if and only if there exists at least one function having different expectation on the two distributions. Consequently we use the maximum discrepancy between function means as the basis of a test statistic.

The Maximum Mean Discrepancy (MMD) can take advantage of the kernel trick, which allows us to apply it not only to vectors, but strings, sequences, graphs, and other common structured data types arising in molecular biology.

Results: We study the practical feasibility of an MMD-based test on three central data integration tasks: Testing cross-platform comparability of microarray data, cancer diagnosis, and data-content based schema matching for two different protein function classification schemas. In all of these experiments, including high-dimensional ones, MMD is very accurate in finding samples that were generated from the same distribution, and outperforms its best competitors.

Conclusions: We have defined a novel statistical test of whether two samples are from the same distribution, compatible with both multivariate and structured data, that is fast, easy to implement, and works well, as confirmed by our experiments.

Availability: http://www.dbs.ifi.lmu.de/~borgward/MMD

Contact: kb@dbs.ifi.lmu.de

1 INTRODUCTION

1.1 Data integration in bioinformatics

The ultimate need for bioinformatics is founded on the wealth of data generated by modern molecular biology. The purpose of bioinformatics is to structure and analyze this data. A central preprocessing step is the integration of datasets that were generated by different laboratories and techniques. If we know how to combine data produced in different labs, we can exploit the results jointly, not only individually. In some cases, the larger datasets thus constructed may support biologically relevant conclusions which were not possible using the original smaller datasets, a hypothetical example being the problem of reliable gene selection from high-dimensional small microarray datasets.

*To whom correspondence should be addressed.

1.2. Distribution testing in data integration

The questions arising in data integration essentially boil down to the following problem of distribution testing: Were two samples X and Y generated by the same distribution? In data integration terms, are these two samples part of the same larger dataset, or should these data be treated as originating from two different sources?

This is a fundamental question when two laboratories are studying the same biological subject. If they use identical techniques on identical subjects but obtain results that are not generated by the same distribution, then this might indicate that there is a difference in the way they generate data, and that their results should not be integrated directly. If the data were integrated without recalibration, differences or patterns within the joint data might be caused by experimental discrepancies between laboratories, rather than by biological processes.

As microarray data are produced by a multitude of different platforms, techniques and laboratories, they are the most prominent data source in bioinformatics for which distribution testing is indispensable. Recently, Marshall (2004) gave an extremely negative picture of cross-platform comparability—and hence the reliability and reproducibility—of microarray results, due to the various platforms and data analysis methods employed (Shi *et al*., 2005). It is therefore crucial for bioinformatics to develop computational methods that allow us to determine whether results achieved across platforms are comparable. In this article, we present a novel statistical test to tackle this problem.

What distinguishes bioinformatics is that it has produced a wealth of complex data, from protein sequences to protein interaction networks, i.e. from strings to graphs. Consequently any practically relevant distribution test needs to be *easily* applicable in all these cases. To the best of our knowledge, the statistical test proposed in our paper is the first method that can handle this wide range of different domains.

To summarize our goals, we will present a novel statistical test for differences in distribution, based on the Maximum Mean Discrepancy (MMD). We will show that it can take advantage of the kernel trick. Hence it is applicable to all data types, from high-dimensional vectors to strings and graphs, arising in bioinformatics. In experiments, we will apply this test to microarray cross-platform comparability testing and cancer diagnosis. Furthermore, we will show how to perform schema matching on complex data by considering a data integration problem on two molecular graph datasets.

Outline of this article In Section 2, we present MMD and its properties. In Section 3, we test the applicability of MMD in cross-platform microarray comparability analysis and cancer diagnosis, and evaluate it on a schema matching problem. We discuss our findings in Section 4.

2 MMD AND THE TWO-SAMPLE PROBLEM

In statistics, the central question of data integration described above is often referred to as the *two-sample* or *homogeneity problem*. The principle underlying the maximum mean discrepancy is that we want to find a function that assumes different expectations on two different distributions. The hope then is that if we evaluate this function on empirical samples from the distributions, it will tell us whether the distributions they have been drawn from are likely to differ. This leads to the following statistic, which is closely related to a proposal by [Fortet and Mourier (1953)]. Here and below, $\mathcal{X}$ denotes our input domain and is assumed to be a nonempty compact set.

DEFINITION 2.1. *Let $\mathcal{F}$ be a class of functions $f:\mathcal{X}\to\mathbb{R}$. Let p and q be Borel probability distributions, and let $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_n)$ be samples composed of independent and identically distributed observations drawn from p and q, respectively. We define the maximum mean discrepancy (MMD) and its empirical estimate as*

$$\mathrm{MMD}[\mathcal{F}, p, q] := \sup_{f\in\mathrm{F}} \left(\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(y)]\right)$$

$$\mathrm{MMD}[\mathcal{F}, X, Y] := \sup_{f\in\mathrm{F}} \left(\frac{1}{m}\sum_{i=1}^{m} f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(y_i)\right)$$

Intuitively it is clear that if $\mathcal{F}$ is 'rich enough', MMD $[\mathcal{F}, p, q]$ will vanish if and only if $p = q$. Too rich an $\mathcal{F}$, however, will result in a statistic that differs significantly from zero for most finite samples X, Y. For instance, if $\mathcal{F}$ is the class of *all* real valued functions on $\mathcal{X}$, and if X and Y are disjoint, then it is trivial to construct arbitrarily large values of MMD$[\mathcal{F}, X, Y]$, for instance by ensuring that $f|_X$ is large and $f|_Y = 0$. This phenomenon of *overfitting* can be avoided by placing restrictions on the function class. That said, these restrictions ought not to prevent the MMD from detecting differences between p and q when these are legitimately to be found. As we shall see, one way to accomplish this tradeoff is by choosing $\mathcal{F}$ to be the unit ball in a universal reproducing kernel Hilbert space, RKHS for short.

We will propose a test of $p = q$, based on an unbiased variant of MMD $[\mathcal{F}, X, Y]$[1] which relies on the asymptotic Gaussianity of this test statistic and on the guaranteed rapid convergence to this asymptotic regime. Thus, the performance guarantees provided by the test apply in the case of a large sample size. The test has a computational cost of $O((m + n)^2)$, although randomization techniques could be employed to reduce the cost to essentially linear time-complexity (at the expense of a somewhat reduced sensitivity).

2.1 MMD for kernel function classes

We now introduce a class of functions for which MMD may easily be computed, while retaining the ability to detect all discrepancies between p and q without making any simplifying assumptions. To this end, let $\mathcal{H}$ be a complete inner product space (i.e., a Hilbert space) of functions $f:\mathcal{X}\to\mathbb{R}$, where $\mathcal{X}$ is a nonempty compact set. Then $\mathcal{H}$ is termed a reproducing kernel Hilbert space if for all $x\in\mathcal{X}$, the linear point evaluation functional mapping $f\to f(x)$ exists and is continuous. In this case, $f(x)$ can be expressed as an *inner product* via

$$f(x) = \langle f\phi(x)\rangle_{\mathrm{H}} \tag{1}$$

where $\phi:\mathcal{X}\to\mathcal{H}$ is known as the *feature space map* from x to $\mathcal{H}$. Moreover, the inner product between two feature maps is called the *(positive definite) kernel*, $k(x, x') := \langle\phi(x), \phi(x')\rangle_{\mathrm{H}}$. Of particular interest are cases where we have an analytic expression for k that can be computed quickly, despite $\mathcal{H}$ being high- or even infinite-dimensional. An example of an infinite-dimensional $\mathcal{H}$ is that corresponding to the Gaussian kernel $k(x, x') = \exp(-\|x - x'\|^2/(2\sigma^2))$.

We will consider universal reproducing kernel Hilbert spaces in the sense defined by Steinwart (2002). Although we do not go into technical detail here, we are guaranteed that RKHSs based on Gaussian kernels are universal, as are string kernels (Section 2.3). See also (Schölkopf *et al*., 2004) for an extensive list of further kernels.

When $\mathcal{F}$ is the unit ball in a universal RKHS, the following theorem (Smola *et al*., 2006) guarantees that MMD$[\mathcal{F}, p, q]$ will detect any discrepancy between p and q.

THEOREM 2.2. *Let p, q be Borel probability measures on $\mathcal{X}$ a compact subset of a metric space, and let $\mathcal{H}$ be a universal reproducing kernel Hilbert space with unit ball $\mathcal{F}$. Then MMD$[\mathcal{F}, p, q] = 0$ if and only if $p = q$.*

Moreover, denote by $\mu_p := \mathbf{E}_p[\phi(x)]$ the expectation of $\phi(x)$ in feature space (assuming that it exists).[2] *Then one may rewrite MMD as*

$$\mathrm{MMD}[\mathcal{F}, p, q] = \|\mu_p - \mu_q\|_{\mathrm{H}}.$$

The main ideas for the proof can be summarized as follows. It is known from probability theory (Dudley, 2002, Lemma 9.3.2) that under the stated conditions, a sufficient condition for $p = q$ is that for all continuous functions f, we have $\int f\,dp = \int f\,dq$. Such functions f, however, can be arbitrarily well approximated using functions in a universal RKHS (Steinwart, 2002). For the second part of the result, observe that due to (1), we may rewrite the MMD as

$$\begin{aligned}\mathrm{MMD}[\mathcal{F}, p, q] &= \sup_{\|f\|_{\mathcal{H}}\le 1} \mathbf{E}_p[f(x)] - \mathbf{E}_q[f(y)] \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \mathbf{E}_p[\langle\phi(x), f\rangle_{\mathcal{H}}] - \mathbf{E}_q[\langle\phi(y), f\rangle_{\mathcal{H}}] \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \langle\mu_p - \mu_q, f\rangle_{\mathcal{H}} = \|\mu_p - \mu_q\|_{\mathcal{H}}.\end{aligned}$$

The finite sample computation of MMD is greatly simplified by (2), as shown in the corollary below:

COROLLARY 2.3. *Under the assumptions of theorem* 2.2 *the following is an unbiased estimator of* MMD$^2[\mathcal{F}, p, q]$:

$$\mathrm{MMD}^2[\mathcal{F}, X, Y] = \frac{1}{m(m-1)}\sum_{i\neq j}^{m} k(x_i, x_j) + \frac{1}{n(n-1)}\sum_{i\neq j}^{n} k(y_i, y_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i, y_j).$$

[1] Note that MMD$[\mathcal{F}, X, Y]$ as defined above is biased: even when $p = q$, it will tend to give strictly positive results for finite sample sizes.

[2] A sufficient condition for this is $\|\mu p\|_{\mathrm{H}}^2 < \infty$, which is rearranged as $\mathbf{E}_p[k(x, x')] < \infty$, where x and x' are independent random variables drawn according to p.

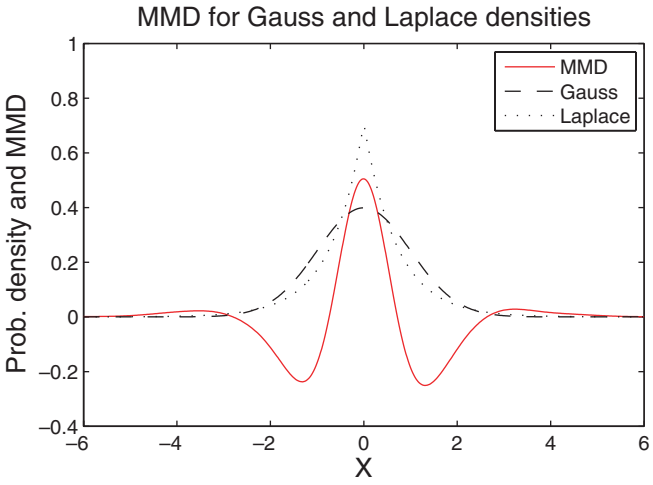


Fig. 1. Illustration of the function maximizing the mean discrepancy in the case where a Gaussian is being compared with a Laplace distribution. Both distributions have zero mean and unit variance. The maximizer of the MMD has been scaled for plotting purposes, and was computed empirically on the basis of 2×10^4 samples, using a Gaussian kernel with $\sigma = 0.5$.

Proof

We compute

$$\begin{aligned}\mathrm{MMD}^2[\mathcal{F}, p, q] &:= \langle \mu_p - \mu_q, \mu_p - \mu_q \rangle_{\mathcal{H}} \\ &= \langle \mu_p, \mu_p \rangle_{\mathcal{H}} + \langle \mu_q, \mu_q \rangle_{\mathcal{H}} - 2\langle \mu_p, \mu_q \rangle_{\mathcal{H}} \\ &= \mathbf{E}_p \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} + \mathbf{E}_q \langle \phi(y), \phi(y') \rangle_{\mathcal{H}} \\ &\quad - 2\mathbf{E}_{p,q} \langle \phi(x), \phi(y) \rangle_{\mathcal{H}},\end{aligned}$$

where x' is a random variable independent of x with distribution p, and y' is a random variable independent of y with distribution q. The proof is completed by applying $\langle \phi(x), \phi(x') \rangle_{\mathrm{H}} = k(x, x')$, and replacing the expectations with their empirical counterparts.

We illustrate the behavior of MMD in Figure 1 using a one-dimensional example: the data X and Y are generated from distributions p and q with equal means and variances, however p is Gaussian and q is Laplacian. For the application of MMD we pick $\mathcal{H}$ to be an RKHS using the Gaussian kernel. We observe that the function f that witnesses the MMD (in other words, the function maximizing the mean discrepancy) is smooth, positive where the Laplace density exceeds the Gaussian density (at the center and tails), and negative where the Gaussian density is larger. Moreover, the magnitude of f is a direct reflection of the amount by which one density exceeds the other, insofar as the smoothness constraint permits it.[3]

Although the expression of $\mathrm{MMD}^2(\mathcal{F}, X, Y)$ in Corollary 2.3 is the minimum variance unbiased estimate (Serfling, 1980), a more tractable unbiased expression can be found in the case where $m = n$, with a slightly higher variance (the distinction is in practice irrelevant, since the terms that differ decay much faster than the variance). It is obtained by dropping the cross-terms $i = j$ from the sum over $k(x_i, y_i)$:

LEMMA 2.4. *Assuming the samples X and Y both have size m, define $z_i = (x_i, y_j)$, and let*

$$h(z_i, z_j) := k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_j, y_i).$$

An unbiased estimate of $\mathrm{MMD}^2[\mathcal{F}, p, q]$ *is given by*

$$\mathrm{MMD}^2[\mathcal{F}, X, Y] := \frac{1}{m(m-1)} \sum_{i \neq j}^{m} h(z_i, z_j).$$

Note that with some abuse of notation we used the *same symbol* as in Corollary 2.3 for a slightly different estimator. However there should be no ambiguity in that we use only the present version for the remainder of the paper.

An important property of the new statistic is that its kernel $h(z_i, z_j)$ is a positive definite kernel in its own right, since

$$h(z_i, z_j) = \langle \phi(x_i) - \phi(y_i), \phi(x_j) - \phi(y_j) \rangle.$$

Thus $z = (x, y) \to \phi(x) - \phi(y)$ is a valid feature map for h. This gives another interpretation of MMD: it is the expected inner product between vectors obtained by connecting a point from one distribution to a point from the other. For detailed discussions of the problem of defining kernels between distributions and sets, see (Cuturi *et al*., 2005; Hein and Bousquet, 2005).

2.2 MMD tests

We now propose a two-sample test based on the asymptotic distribution of an unbiased estimate of MMD^2, which applies in the case where $\mathcal{F}$ is a unit ball in a RKHS, and $m = n$. This uses the following theorem, due to Hoeffding (1948). See also Serfling (1980, Section 5.5.1). For a proof and further details see Smola *et al*. (2006).

THEOREM 2.5. *Let z_i and $h(z_i, z_j)$ be specified as in Definition* 2.4 *and assume that* $\mathbf{E}_{p,q}[\mathrm{MMD}^4[\mathcal{F}, X, Y]] < \infty$. *Then for $m \to \infty$, the statistic* $\mathrm{MMD}^2(\mathcal{F}, X, Y)$ *converges in distribution to a Gaussian with mean* $\mathrm{MMD}^2[\mathcal{F}, p, q]$ *and variance*

$$\sigma^2_{\mathrm{MMD}} = \frac{2^2}{m}\left(\mathbf{E}_z[(\mathbf{E}_{z'} h(z, z'))^2] - [\mathbf{E}_{z,z'}(h(z, z'))]^2\right).$$

The convergence to the normal occurs rapidly: according to Serfling (1980, Theorem B,p. 193), the CDF of the U-statistic converges uniformly to the asymptotic CDF at rate $1/\sqrt{m}$.

Our goal is to test whether the above normal distribution has zero mean (the null hypothesis), as opposed to a mean that is positive. Since we need not concern ourselves with negative deviations from the mean ($\mathrm{MMD}[\mathcal{F}, p, q] \geq 0$ may never become negative), it suffices to test whether $\mathrm{MMD}^2[\mathcal{F}, X, Y] \leq \varepsilon$ for some threshold ε. Thus, we obtain the two-sample test below as a corollary to Theorem 2.5, following the principles outlined by Casella and Berger (2002, Section 10.3.2).

[3] One may show that the maximizer of $\mathrm{MMD}[\mathcal{F}, p, q]$ is given by $f(x) = \langle \mu_p - \mu_q, \phi(x) \rangle$. The same holds true for the maximizer of the empirical quantity, with the means being replaced by empirical means. See (Smola *et al*., 2006) for further details and a proof.

Algorithm 1 MMD test using asymptotic normality

Input: positive definite kernel k, level of test $\alpha \in (0, 1)$, samples X and Y of size m drawn from p and q respectively

$\text{MMD}^2 \leftarrow 0$ and $\sigma^2 \leftarrow 0$

for $i = 1$ to m **do**

$t \leftarrow 0$

for $j = 1$ to m **do**

if $j \neq i$ **then**

$t \leftarrow t + k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_i, y_i)$

end if

end for

$\text{MMD}^2 \leftarrow \text{MMD}^2 + \frac{1}{m(m-1)} t$ and $\sigma^2 \leftarrow \sigma^2 + t^2$

end for

$\sigma^2 \leftarrow \frac{4}{(m^2(\text{m}-1)^2)}\sigma^2 - \frac{4}{m}(MMD^2)^2$

$\epsilon \leftarrow \sqrt{2\sigma^2}$ erfinv $(1-2\alpha)$

Output: If $\text{MMD}^2 \leq \epsilon$ return $p = q$ accepted. Otherwise return $p = q$ rejected.

COROLLARY 2.6. *A test of the null hypothesis $p = q$ with asymptotic size*[4] *α, and asymptotic Type II error zero, has the acceptance region*

$$\text{MMD}^2[\mathcal{F}, X, Y] \leq \hat{\sigma}_{\text{MMD}} z_\alpha$$

where

$$\hat{\sigma}^2_{\text{MMD}} = \frac{4}{m^2(m-1)^2} \sum_{i=1}^{m} \Big(\sum_{j \neq i}^{m} h(z_i, z_j) \Big)^2 - \frac{4}{m} \text{MMD}^4[\mathcal{F}, X, Y]$$

or any empirical estimate of σ_{MMD} that converges in probability. Here z_α satisfies $\Pr(z > z_\alpha) = \alpha$ when $z \sim \mathcal{N}(0,1)$.

It is also of interest to estimate the p-value of the test. We first describe a sample-based heuristic. We draw randomly without replacement from the aggregated data $Z = \{X, Y\}$ to get two new m-samples X^* and Y^*, and compute the test statistic $\text{MMD}^2_*(\mathcal{F}, X^*, Y^*)$ between these new samples (bear in mind that under the null hypothesis $p = q$, this aggregation is over data drawn from a single distribution). We repeat this procedure t times to obtain a set of test statistics under the null hypothesis (conditioned on the observations). We then add the original statistic $\text{MMD}^2(\mathcal{F}, X, Y)$ to this set, and sort the set in ascending order. Define as r the rank of the original test statistic within this ordering. Then our estimated p-value is $p = (t + 1 - r)/(t + 1)$. Alternatively, we can find an upper bound on p using the distribution-free large deviation result of Hoeffding (1963, p. 25) (see Smola *et al.*, 2006, Section 6), which is exact at finite sample sizes. This bound is only tight when m is large, however, and may be too conservative at small sample sizes.

We give the complete pseudocode for the above MMD-based test in Algorithm 1. We emphasize that the computational cost is $O(m^2)$, and that the method is easily parallelized (the kernel matrix can be broken up into submatrices, and the relevant sums computed independently before being combined). In addition, the kernel matrix needs never be stored in memory, but only a running sum must be kept, which makes the analysis of very large data sets feasible. Randomized methods could also be used to speed up the double-loop required for evaluating Algorithm 6, by only computing parts of the sum. This procedure would reduce the quality of the test, however.

Finally, we note that other approaches are also possible in determining the acceptance region of the test. For instance, Smola *et al.* (2006) describe two tests based on large deviation bounds: the first uses Rademacher averages to obtain a bound that explicitly accounts for the variation in the test statistic, the second uses a distribution-independent upper bound on the test statistic variation due to Hoeffding (1963, p. 25). These approaches have the advantage of giving an exact, distribution-free test of level α that holds for finite samples, and not just in the asymptotic regime. In addition, they provide a finite sample upper bound on the p-value, which is again distribution-free. A disadvantage of these approaches is that they require a larger sample size than the test in Corollary 6 before they can detect a given disparity between the distributions p and q, i.e. they have a higher Type II error. For this reason, we do not use these tests in Section 3.

2.3 Universal kernels for discrete data

While many examples of universal kernels on compact subsets of $\mathbb{R}^d$ are known (Steinwart, 2002), little attention has been given to finite domains. It turns out that the issue is considerably easier in this case: the weaker notion of *strict positive definiteness* (kernels inducing nonsingular Gram matrices $(k(x_i, x_j))_{ij}$ for arbitrary sets of distinct points x_i) ensures that every function on a discrete domain $x = \{x_1, \ldots, x_n\}$ lies in the corresponding RKHS (and hence that the kernel is universal). To see this, let $f \in \mathbb{R}^n$ be an arbitrary function on $\mathcal{X}$. Then $\alpha = K^{-1}f$ ensures that the function $f = \sum_j k(.,x_j)$ satisfies $f(x_i) = f_i$ for all i.

It turns out that string kernels fall in this class:

THEOREM 2.7. *Let $\mathcal{X}$ be a finite set of strings, and let $\#_s(x)$ denote the number of times substring s occurs in x. Then any string kernel of the form $k(x, x') = \sum_{s \in X} w_s \#_s(x) \#_s(x')$ with $w_s > 0$ for all $s \in \mathcal{X}$ is strictly positive definite.*

Proof. We will show that the vectors $\{\phi(x) \,|\, x \in \mathcal{X}\}$ obtained by the feature map are linearly independent, implying that all Gram matrices are nonsingular. The feature map is given by $\phi(x) = (\sqrt{w_s}\#_s(x), \sqrt{w_{s'}}\#_{s'}(x), \ldots)$ where we assume for the purpose of the proof that all substrings s are ordered by nondecreasing length. Now for a given set X of size m consider the matrix with columns $\phi(x_1), \ldots, \phi(x_m)$, where the entries in X are assumed to be ordered in the same manner as the substrings (i.e. by nondecreasing length). By construction, the upper triangle of this matrix is zero, with the highest nonzero entry of each row being $\sqrt{w_x}$, which implies linear independence of its rows.

For graphs unfortunately no strictly positive definite kernels exist which are efficiently computable. Note first that it is necessary for strict positive definiteness that $\phi(x)$ be injective, for otherwise we would have $\phi(x) = \phi(x')$ for some $x \neq x'$, implying that the kernel matrix obtained from $X = \{x, x'\}$ is singular. However, as Gärtner *et al.* (2003) show, an injective $\phi(x)$ allows one to match graphs by computing $\|\phi(x) - \phi(x')\|^2 = k(x,x) + k(x',x') - 2k(x,x')$. Graph matching, however, is NP-hard, hence no such

[4] Size and level are defined following Casella and Berger (2002, Section 8.3).

kernel can exist. That said, there exists a number of useful graph kernels. See e.g. (Borgwardt *et al*., 2005) for further details.

2.4 Kernel choice

So far, we have focused on the case of universal kernels. These kernels have various favorable properties, including that

- universal kernels are strictly positive definite, making the kernel matrix invertible and avoiding non-uniqueness in the dual solutions of SVMs,
- Continuous functions on $\mathcal{X}$ can be arbitrarily well approximated (in the $\|\cdot\|_\infty$-norm) using an expansion in terms of universal kernels, and SVMs using universal kernels are consistent in the sense that (subject to certain conditions) their solutions converge to the Bayes optimal solution (Steinwart, 2002).
- MMD using universal kernels is a test for identity of arbitrary Borel probability distributions.

However, note that for instance in pattern recognition, there might well be situations where the best kernel for a given problem is not universal. In fact, the kernel corresponds to the choice of a prior, and thus using a kernel which does *not* afford approximations of arbitrary continuous functions can be very useful—provided that the functions it does approximate are known to be solutions of the given problem.

The situation is similar for MMD. Consider the following example: suppose we knew that the two distributions we are testing are both Gaussians (with unknown mean vectors and covariance matrices). Since the empirical means of products of input variables up to order two are sufficient statistics for the family of Gaussians, we should thus work in an RKHS spanned by products of order up to two—any higher order products contain no information about the underlying Gaussians and can therefore mislead us. It is straightforward to see that for $c > 0$, the polynomial kernel $k(x,x') = (\langle x,x'+c)^2$, with c > 0, does the job: it equals

$$\sum_{i,j=1}^{d} x_i x_j x'_i x'_j + 2c \sum_{i=1}^{d} x_i x'_i + c^2 = \langle \phi(x), \phi(x') \rangle,$$

where $\phi(x) = (c, \sqrt{2c}x_1, \ldots, \pm\sqrt{2c}x_d, x_i x_j \,|\, i,j = 1,\ldots,d)^\top$. If we want to test for differences in higher order moments, we use a higher order kernel[5] $k(x,x') = (\langle x,x\rangle + c)^p$.

Note, however, that this does not tell us how to choose c. With additional prior knowledge, we could further improve the odds of our test working well on small sample sizes. For instance, if we knew that the Gaussians differ mainly in their covariance structures, then we could incorporate this by choosing a small c. If the available prior knowledge is less specific, we could also sum up several MMDs by using summed kernels.

2.5 Related methods

Various empirical methods have been proposed to determine whether two distributions are different. The first test we consider, and the simplest, is a multivariate generalization of the t-test (Hotelling, 1951), which assumes both distributions are multivariate Gaussian with unknown, identical covariance structure. This test is not model-free in the sense of MMD (and the tests described below)—indeed, it is easy to construct examples in which it fails completely (Figure 1).

Two well-established model-free univariate tests are the Kolmogorov-Smirnov statistic and the Wald-Wolfowitz runs test. Both tests are powerful in that the distribution of the test statistic is known independently of p and q for finite sample sizes, under the null hypothesis $p = q$. A generalization of the Wald-Wolfowitz runs test to the multivariate domain was proposed by Friedman and Rafsky (1979). It involves counting the number of edges in the minimum spanning tree over the aggregated data that connect points in X to points in Y. The resulting test relies on the asymptotic normality of the test statistic. The computational cost of this method using Kruskal's algorithm is $O((m+n)^2\log(m+n))$, although more modern methods improve on the $\log(m+n)$ term. Two possible generalizations of the Kolmogorov-Smirnov test to the multivariate case were studied by Bickel (1969); Friedman and Rafsky (1979). The approach of Friedman and Rafsky in this case again requires a minimal spanning tree, and thus has a similar cost to their multivariate runs test.

Hall and Tajvidi (2002) propose to aggregate the data as $Z = \{X,Y\}$, find the j points in Z closest to each point in X for all $j \in \{1,\ldots,m\}$, count how many of these are from Y, and compare this with the number of points expected under the null hypothesis (the procedure is repeated for each point in Y wrt points in X). The test statistic is costly to compute; Hall and Tajvidi (2002) consider only tens of points in their experiments.

Another approach is to use some distance (e.g. L_1 or L_2) between estimates of the densities as a test statistic (Anderson *et al*., 1994; Biau and Gyorfi, 2005), based on the asymptotic distribution of this distance given $p = q$. One problem with the approach of Biau and Gyorfi (2005), however, is that it requires the space to be partitioned into a grid of bins, which becomes difficult or impossible for high dimensional problems (such as those in Section 3).

We now illustrate these tests with a simple example. In Figure 2, we compare several alternatives to the MMD-based test in distinguishing 100 samples taken from each of two normal distributions with unit variance. Results are averaged over a series of Euclidean distances between the means of both distributions, and plotted as a function of increasing dimensionality. The t-test has the highest chance of correctly rejecting the null hypothesis for low dimensions. However, for high dimensions the estimation of the sample covariance matrices is poor due to the limited sample sizes. Note that we do *not* apply the Biau & Györfi test for high dimensionalities, since memory requirements force the number of partitions per dimension to be too low.

MMD performs very well and outperforms all other model-free approaches, namely the multivariate Kolmogorov-Smirnov test (FR Smirnov), the multivariate Wald-Wolfowitz runs test (FR Wolf), and the Biau & Györfi test (Biau). The comparison becomes harder for increasing dimensionality, since the sample size is fixed to 100 random vectors per distribution for all dimensions. Moreover, MMD also yields a very low rejection rate of the null hypothesis, when it is true (see figure legend).

Finally, we mention that the connection between means in RKHSs and distributions has, in a less general setting, been observed before in the field of kernel machines. Schölkopf and Smola (2002) point out that the empirical mean of a set of points

[5] Kernels with infinite-dimensional RKHS can be viewed as a nonparametric generalization where we have infinitely many sufficient statistics.

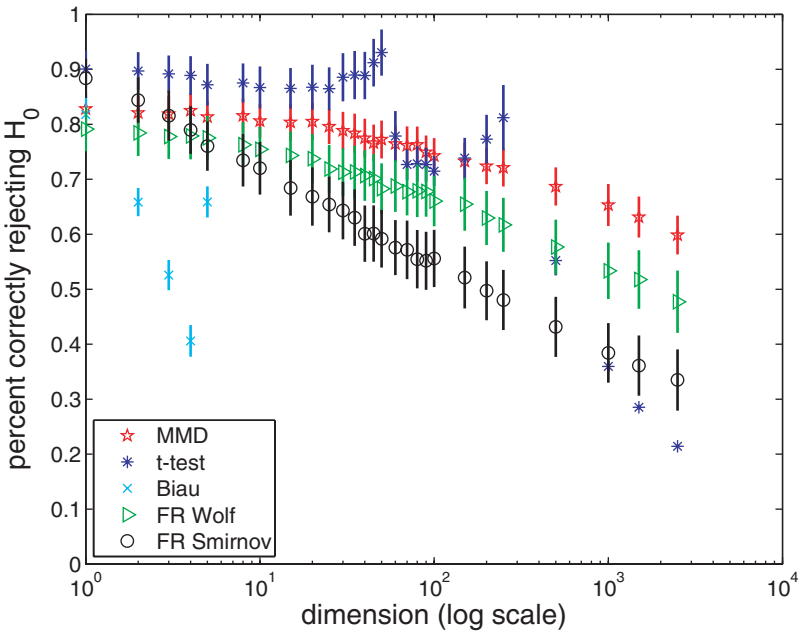


Fig. 2. Test of samples from two normal distributions with different means and unit variance, based on a significance level $\alpha = 0.05$. The cumulative percentage of times the null hypothesis was correctly rejected over the set (0.1,0.2,0.25,0.3,0.4,0.5,0.6,0.7,0.8,1,2,5,10,15) of Euclidean distances between the distribution means, was computed as a function of the dimensionality of the normal distributions. Its average and standard error in 333 repetitions is shown for each of the five tests employed. The sample size was 100 for each distribution. The MMD used a Gaussian kernel, with kernel size σ obtained by maximizing MMD (for σ values within 0.25 and 20) to get the most conservative test. In case of the t-test, a ridge was added to the covariance estimate, to avoid singularity (the ridge was incremented in steps of 0.01 until the 2-norm condition number was below 10). For the Biau test, equal partitions per dimension were used, although this becomes intractable for high dimensions. When samples from distributions with equal mean were compared, the tests wrongly rejected the null hypothesis in the following number of trials out of 8991 (summed over all dimensions in the plot, with 333 runs each): 112 (MMD), 960 (t-test), 379 (FR Wolf), 441 (FR Smirnov). For the Biau test: 4 out of 1665 trials.

in an RKHS can be viewed as a Parzen windows estimate of the density underlying the data; and Shawe-Taylor and Cristianini (2004) propose to use the distance to the mean as a novelty detection criterion, and provide a statistical analysis.

3 EXPERIMENTS

In this section, we present applications of MMD in data integration for bioinformatics, namely microarray cross-platform comparability, cancer (subtype) diagnosis, and schema matching for enzyme protein structures.

3.1 Microarray cross-platform comparability

Experimental scenario Microarrays as a large-scale gene expression observation tool offer a unique possibility for molecular biologists to study gene activity at a cellular level. In recent years, there have been a great number of developments in different microarray platforms, techniques and protocols, advances in these techniques, and biological and medical studies making use of these approaches. As a result, microarray data for a given problem, and the results derived from it (e.g. marker genes for a certain subtype of cancer), may vary greatly (Carter *et al*., 2005), both between labs and platforms. Even for the subsequent step of data processing, e.g. missing value imputation, a large battery of different techniques is available. Consequently, despite an avalanche of microarray data being generated nowadays, it remains to be determined if and how to combine microarray data from different studies on the same biological subject.

Therefore, it is necessary to establish a statistical test of whether two microarray measurements on the same subject, obtained by two different labs or on two different platforms, can be regarded as comparable and can be used for joint data analysis. We define such a test using MMD as a statistic: if an MMD-based test rejects the null hypothesis that the microarray measurements are generated from the same distribution, then we deem them not comparable.

We test this approach on published microarray datasets from two different platforms. If our criterion is useful in practice and able to detect the limited cross-platform comparability of microarray data, then MMD should judge microarray data achieved on different platforms as being less often comparable than those found on the same platform.

Data For our first experiment, we obtained 2 datasets from Warnat *et al*. (2005), from two studies on breast cancer by Gruvberger *et al*. (2001) and West *et al*. (2001). Both comprise gene expression levels for a common set of 2,166 genes. Different microarray platforms were used in these studies: while Gruvberger *et al*. (2001) achieved their results on a c-DNA platform, West *et al*. (2001) utilized oligonucleotide microarrays.

We tried to find out via MMD if there is any statistically significant difference between the microarray results achieved on these different platforms. Samples were scaled to zero mean and unit variance beforehand, although not for the t-test. We compared

Table 1. Microarray cross-platform comparability

Platforms	H_0	MMD	t-test	FR Wolf	FR Smirnov
Same	accepted	100	100	93	95
Same	rejected	0	0	7	5
Different	accepted	0	95	0	29
Different	rejected	100	5	100	71

Cross-platform comparability tests on microarray level for cDNA and oligonucleotide platforms. Repetitions 100, sample size (each) 25, dimension of sample vectors: 2,116

Table 2. Cancer diagnosis

Health status	H_0	MMD	t-test	FR Wolf	FR Smirnov
Same	accepted	100	100	97	98
Same	rejected	0	0	3	2
Different	accepted	0	100	0	38
Different	rejected	100	0	100	62

Comparing samples from normal and prostate tumor tissues (Singh *et al*., 2002). H_0 is hypothesis that $p = q$. Repetitions 100, sample size (each) 25, dimension of sample vectors: 12,600

the MMD results to the multivariate t-test and the Friedman-Rafsky multivariate Kolmogorov-Smirnov and Wald-Wolfowitz tests (denoted Smirnov and Wolf, respectively). The high dimensionality of this problem, as well as of the experiments below, prevents a comparison with the Biau-Györfi test.

We chose $\alpha = 0.05$ as the level of significance for all tests. A Gaussian kernel was employed for MMD, with $\sigma = 20$. We obtained an average performance over 100 distribution tests using 50 microarray measurements from different platforms (X being 25 cDNA measurements and Y being 25 oligonucleotide measurements), and 100 distribution tests with data from 50 microarray measurements taken from only one of the two platforms. For each test, the studies were randomly selected without replacement from the relevant measurement pools. We repeated this experiment for MMD and each of the competing methods.

Results Results are reported in Table 1 , showing the number of times MMD and the other three methods deemed two samples as originating from the same distribution, on data from both identical and dissimilar platforms. In the majority of repetitions, both MMD and the Friedman-Rafsky tests recognize correctly whether two samples were generated on the same platform or not. However MMD is the only test that makes no Type I or Type II errors in all repeats of the experiment. While the FR Wolf test has no false negatives when the samples are from different platforms, it finds occasional false positives when the samples arise from the same platform. The FR Smirnov test has a slightly reduced Type I error rate compared with the FR Wolf test, but at the expense of a much larger Type II rate. Finally, the t-test appears unable to distinguish differences in platform, which is unsurprising given the high dimensionality of the data. As inter-platform comparability of microarray data is reported to be modest in many recent publications (van Ruissen *et al*., 2005; Carter *et al*., 2005; Stec *et al*., 2005), MMD is very successful in detecting these differences in our experiments. We also note that our sample sizes are relatively small, which makes problematic the assumption of both the MMD and Friedman-Rafsky tests that the associated statistic has an asymptotic distribution (this remark also holds for the experiments in the next section). That said, this approximation appears reasonable for the tasks we address, in the light of our results.

3.2 Cancer and tumor subtype diagnosis

Experimental scenario Besides microarray cross-platform comparability, it is interesting to examine whether MMD can distinguish between the gene expression profiles of groups of people who are respectively healthy or ill, or who suffer from different subtypes of a particular cancer. Alternatively, as in the previous experiment, MMD can be employed to determine whether we should integrate two sets of observations (which might arise from different subtypes of a cancer) into one joint set, or if we should treat them as distinct classes.

When using MMD for cancer diagnosis, we test whether the microarray data at hand contain a significant level of difference between ill and healthy patients. Conversely, when looking at cancer (or tumor) subtypes, MMD indicates whether two subtypes of cancer should be considered independently when designing a computational predictor of cancer, or if they can be assigned to one common super-class. In terms of classification methods, MMD can be used to choose whether binary (cancer/healthy) or multi-class (healthy, cancer subtype 1, . . . , cancer subtype n) classification will be more accurate when developing a diagnosis tool.

Data For our second microarray experiment, we obtained datasets from two cancer microarray studies. The first, by Singh *et al*. (2002), is a dataset of gene expression profiles from 52 prostate tumor and 50 normal, non-tumor samples. The second, by Monti *et al*. (2005), consists of microarray data from diffuse large B-cell lymphoma samples. In particular, we are interested in cancer diagnosis on the data of Singh *et al*. (2002), and tumor subtype diagnosis on the data of Monti *et al*. (2005). We again normalized each data sample to zero mean and unit variance, besides for the t-test.

Cancer diagnosis

We examine whether MMD can distinguish between normal and tumor tissues, using the microarray data from the prostate cancer study by Singh *et al*. (2002). Again, α was set to 0.05. Randomly choosing 100 pairs of 25 healthy and 25 cancer patients' gene expression profiles, we used MMD to test the null hypothesis that both samples were generated by the same distribution. We then did the same test for 100 randomly chosen pairs of samples of size 25, both drawn from the same tissue type (healthy or tumor). For all 200 pairs of samples, we compared our results to those of the multivariate t-test and both Friedman-Rafsky tests (Wolf and Smirnov).

Results Results are reported in Table 2. Both MMD and the Friedman-Rafsky tests are in agreement that there is a large difference between samples from cancer patients and healthy patients, and little difference within a particular class. We again see that both MMD and FR Wolf make no Type II errors, but that only MMD makes no Type I errors; and that FR Smirnov has a much higher Type II error rate than FR Wolf (while making one fewer Type I errors).

Table 3. Tumor subtype tests

Subtype	H_0	MMD	t-test	FR Wolf	FR Smirnov
Same	accepted	100	100	95	96
Same	rejected	0	0	5	4
Different	accepted	0	100	0	22
Different	rejected	100	0	100	78

Comparing samples from different and identical tumor subtypes of lymphoma (Monti *et al*., 2005). H_0 is hypothesis that $p = q$. Repetitions 100, sample size (each) 25, dimension of sample vectors: 2,118.

Table 4. Data-content based schema matching

Test	EC 1	EC 2	EC 3	EC 4	EC 5	EC 6
EC 1	0	50	45	50	50	50
EC 2	50	0	50	50	50	50
EC 3	48	50	0	50	50	50
EC 4	50	50	50	0	50	50
EC 5	50	50	50	50	0	50
EC 6	50	50	50	50	50	0

Data-content based schema matching for $\alpha = 0.01$. Numbers indicate how often null hypothesis ($p = q$) was rejected.

Tumor subtype diagnosis

We performed the same experiment as above for tumor subtype diagnosis on data from Monti *et al*. (2005). We are interested in whether MMD is able to distinguish betweeen lymphoma of two subtypes: ''oxidative phosphorylation'' and ''B-cell receptor/proliferation''.

Results We report results in Table 3. As in the previous experiment, both MMD and the Friedman-Rafsky tests prefer to reject the null hypothesis that both samples are generated by the same distribution, when the lymphoma subtypes are different. In other words, all three tests succeed in finding discrepancies between samples from different tumor subtypes in this case. This is consistent with previous results by Monti *et al*. (2005) who discovered these different lymphoma subtypes by using a combination of several clustering algorithms. Hence MMD confirms the existence of these subtypes in our experiment. Comparing the performance of the various tests gives results consistent with the previous two experiments: MMD and FR Wolf do not make any Type II errors, but only MMD has no Type I errors; and FR Smirnov has a much worse Type II error rate than FR Wolf, but makes one fewer Type I errors.

3.3 Schema matching on molecular graph data

Experimental scenario Classifying biological data into ontologies or taxonomies is the central step in structuring and organizing the data. However, different studies may use different ontologies, resulting in the need to find correspondences between two ontologies. We employ MMD to discover matching terms in two ontologies using the data entries associated with these terms.

We study the following scenario: Two researchers have each dealt with 300 enzyme protein structures. These two sets of 300 proteins are disjunct, i.e. there is no protein studied by both researchers. They have assigned the proteins to six different classes according to their enzyme activity. However, both have used different protein function classification schemas for these six classes, and are not sure which pairs of classes correspond.

To find corresponding classes, the MMD can be employed. We obtained 600 proteins modeled as graphs from Borgwardt *et al*. (2005), and randomly split these into two subsets A and B of 300 proteins each, such that 50 enzymes in each subset belong to each one of the six EC top level classes. We then computed MMD for all pairs of EC classes from subset A and subset B to check if the null hypothesis is rejected or accepted. To compute the MMD, we employed the protein random walk kernel for protein graphs, following Borgwardt *et al*. (2005). We compared all pairs of classes via MMD, and repeated the experiment 50 times.

Results For a significance level of $\alpha = 0.05$, MMD rejected the null hypothesis that both samples are from the same distribution whenever enzymes from two different EC classes were compared. When enzymes from the same EC classes were compared, MMD accepted the null hypothesis. MMD thus achieves error-free data-based schema matching here.

We checked whether the same good results were found for a higher significance level of $\alpha = 0.01$. We report results in Table 4. This time, in 7 comparisons out of 1800 comparisons the null hypothesis is incorrectly accepted, whereas in all other cases, the correct decision is taken. Hence even for the high significance level of $\alpha = 0.01$ MMD is very accurate.

In addition to these promising results, note that although we consider the basic case of 1:1 correspondence between classes in our experiment, the fact that MMD uses the kernel trick allows for even more powerful approaches to data-content based schema matching. As kernels are closed under addition and pointwise multiplication, we can test complex correspondences between different classes as well, where one class in schema A corresponds to a combination of classes in schema B. Schema matching for complex correspondences via MMD is a topic of current research.

4 DISCUSSION AND CONCLUSIONS

In this paper, we have presented, to the best of our knowledge, the first principled statistical test for distribution testing and data integration of structured objects, using the Maximum Mean Discrepancy (MMD) as a test statistic. MMD makes use of kernels, and hence is not limited to vector data. As a consequence, MMD is not only applicable to a wide range of problems in molecular biology, but also to common data types in bioinformatics, such as strings and graphs. Kernels for biological data, which have previously been used in classification tasks, can now be employed for distribution testing. Amongst others, these include kernels on protein sequences, protein structures, and microarray time series (Schölkopf *et al*., 2004).

MMD is easy to implement, memory-efficient, and fast to compute. In all of our experiments, it outperformed competing methods (provided the latter were applicable at all, i.e., on vectorial data). We applied our MMD-based test to microarray cross-platform comparability, cancer diagnosis, and data-content based schema matching.

We believe that MMD could also be employed to validate computational simulations of biological processes. If wetlab experiments and simulations generate results and predictions that MMD deems comparable, it is likely that the simulator has produced realistic predictions. This validation procedure will become increasingly relevant as more model-based simulations of microarray data become available (den Bulcke *et al*., 2006).

MMD could also be used for keyplayer gene selection from microarray data. This type of feature selection could be employed to find genes that are involved in a cancer outbreak when looking at gene expression profiles from healthy and cancer patients. MMD would be applied to subsets of genes from two classes of microarrays to find the subset that maximizes the probability that the two classes arise from different distributions. These genes should be studied experimentally in more detail. If, however, MMD cannot find any subset of genes that results in significant differences between healthy and cancer patients, then this might serve as an indicator that the microarrays did not contain the essential genes involved in cancer progress.

ACKNOWLEDGEMENTS

The authors are grateful to Patrick Warnat (DKFZ, Heidelberg) for providing datasets for one of our MMD experiments, and to Olivier Bousquet and Matthias Hein for helpful discussions. This work was supported in part by National ICT Australia and by the German Ministry for Education, Science, Research and Technology (BMBF) under grant no. 031U112F within the BFAM (Bioinformatics for the Functional Analysis of Mammalian Genomes) project which is part of the German Genome Analysis Network (NGFN). National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council. This work was supported in part by the Austrian Science Fund Fonds zur Förderung der Wissenschaftlichen Forschung (FWF), project # S9102-N04, and the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

REFERENCES

N. Anderson, P. Hall, and D. Titterington. Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates. *Journal of Multivariate Analysis*, 50:41–54, 1994.

G. Biau and L. Gyorfi. On the asymptotic properties of a nonparametric l_1-test statistic of homogeneity. *IEEE Transactions on Information Theory*, 51(11):3965–3973, 2005.

P. Bickel. A distribution free version of the smirnov two sample test in the p-variate case. *The Annals of Mathematical Statistics*, 40(1):1–23, 1969.

K. M. Borgwardt, C. S. Ong, S. Schonauer, S. V. N. Vishwanathan, A. J. Smola, and H. P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21 (Suppl 1):i47–i56, Jun 2005.

S. L. Carter, A. C. Eklund, B. H. Mecham, I. S. Kohane, and Z. Szallasi. Redefinition of affymetrix probe sets by sequence overlap with cdna microarray probes reduces cross-platform inconsistencies in cancer-associated gene expression measurements. *BMC Bioinformatics*, 6(1):107, 2005.

G. Casella and R. Berger. *Statistical Inference*. Duxbury, Pacific Grove, CA, 2nd edition, 2002.

M. Cuturi, K. Fukumizu, and J.-P. Vert. Semigroup kernels on measures. *JMLR*, 6: 1169–1198, 2005.

T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7(1):43, 2006.

R. M. Dudley. *Real analysis and probability*. Cambridge University Press, Cambridge, UK, 2002.

R. Fortet and E. Mourier. Convergence de la réparation empirique vers la réparation théorique. *Ann. Scient. École Norm. Sup*., 70:266–285, 1953.

J. Friedman and L. Rafsky. Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *The Annals of Statistics*, 7(4):697–717, 1979.

T. Gärtner, P.A. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In B. Schölkopf and M. K. Warmuth, editors, *Proc. Annual Conf. Computational Learning Theory*. Springer, 2003.

S. Gruvberger, M. Ringner, Y. Chen, S. Panavally, L. H. Saal, A. Borg, M. Ferno, C. Peterson, and P. S. Meltzer. Estrogen receptor status in breast cancer is associated with remarkably distinct gene expression patterns. *Cancer Res*, 61 (16): 5979–5984, Aug 2001.

P. Hall and N. Tajvidi. Permutation tests for equality of distributions in high-dimensional settings. *Biometrika*, 89(2):359–374, 2002.

M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In Z. Ghahramani and R. Cowell, editors, *Proc. of AI & Statistics*, volume 10, 2005.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948.

H. Hotelling. A generalized t test and measure of multivariate dispersion. *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 23–41, 1951.

E. Marshall. Getting the noise out of gene arrays. *Science*, 306(5696):630–631, Oct 2004.

S. Monti, K. J. Savage, J. L. Kutok, F. Feuerhake, P. Kurtin, M. Mihm, B. Wu, L. Pasqualucci, D. Neuberg, R. C. Aguiar, P. Dal Cin, C. Ladd, G. S. Pinkus, G. Salles, N. L. Harris, R. Dalla-Favera, T. M. Habermann, J. C. Aster, T. R. Golub, and M. A. Shipp. Molecular profiling of diffuse large b-cell lymphoma identifies robust subtypes including one characterized by host inflammatory response. *Blood*, 105(5):1851–1861, Mar 2005.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA, 2004.

R. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley, New York, 1980.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

L. Shi, W. Tong, H. Fang, U. Scherf, J. Han, R. K. Puri, F. W. Frueh, F. M. Goodsaid, L. Guo, Z. Su, T. Han, J. C. Fuscoe, Z. A. Xu, T. A. Patterson, H. Hong, Q. Xie, R. G. Perkins, J. J. Chen, and D. A. Casciano. Cross-platform comparability of microarray technology: intra-platform consistency and appropriate data analysis procedures are essential. *BMC Bioinformatics*, 6 Suppl 2:S12, Jul 2005.

D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1 (2): 203–209, Mar 2002.

A.J. Smola, A. Gretton, and K. Borgwardt. Maximum mean discrepancy. Technical Report NICTA-SML-06-001, National ICT Australia, 2006. URL http://sml.nicta.com.au/smola/papers/SmoGreBor06tr.pdf.

J. Stec, J. Wang, K. Coombes, M. Ayers, S. Hoersch, D. L. Gold, J. S. Ross, K. R. Hess, S. Tirrell, G. Linette, G. N. Hortobagyi, W. Fraser Symmans, and L. Pusztai. Comparison of the predictive accuracy of dna array-based multigene classifiers across cdna arrays and affymetrix genechips. *J Mol Diagn*, 7(3):357–367, Aug 2005.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2002.

F. van Ruissen, J. M. Ruijter, G. J. Schaaf, L. Asgharnegad, D. A. Zwijnenburg, M. Kool, and F. Baas. Evaluation of the similarity of gene expression data estimated with sage and affymetrix genechips. *BMC Genomics*, 6:91, Jun 2005.

P. Warnat, R. Eils, and B. Brors. Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, 6: 265, Nov 2005.

M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. r. Olson JA, J. R. Marks, and J. R. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc Natl Acad Sci USA*, 98(20):11462–11467, Sep 2001.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e58–e65
doi:10.1093/bioinformatics/btl212

ACIAP, Autonomous hierarchical agglomerative Cluster Analysis based protocol to partition conformational datasets

Giovanni Bottegoni[1], Walter Rocchia[2], Maurizio Recanatini[1] and Andrea Cavalli[1,*]

[1]Department of Pharmaceutical Sciences, University of Bologna, Via Belmeloro 6, I-40126, Bologna, Italy and [2]NEST CNR-INFH, Scuola Normale Superiore of Pisa, Piazza dei Cavalieri, 7, I-56126, Pisa, Italy

ABSTRACT

Motivation: Sampling the conformational space is a fundamental step for both ligand- and structure-based drug design. However, the rational organization of different molecular conformations still remains a challenge. In fact, for drug design applications, the sampling process provides a redundant conformation set whose thorough analysis can be intensive, or even prohibitive. We propose a statistical approach based on cluster analysis aimed at rationalizing the output of methods such as Monte Carlo, genetic, and reconstruction algorithms. Although some software already implements clustering procedures, at present, a universally accepted protocol is still missing.

Results: We integrated hierarchical agglomerative cluster analysis with a clusterability assessment method and a user independent cutting rule, to form a global protocol that we implemented in a MATLAB metalanguage program (ACIAP). We tested it on the conformational space of a quite diverse set of drugs generated via Metropolis Monte Carlo simulation, and on the poses we obtained by reiterated docking runs performed by four widespread programs. In our tests, ACIAP proved to remarkably reduce the dimensionality of the original datasets at a negligible computational cost. Moreover, when applied to the outcomes of many docking programs together, it was able to point to the crystallographic pose.

Availability: ACIAP is available at the ''ACIAP'' section of the website http://www.scfarm.unibo.it.

Contact: E-mail: andrea.cavalli@unibo.it.

Supplementary Information: The complete series of ACIAP results is available in the ''services'' section of the website http://www.scfarm.unibo.it.

1 INTRODUCTION

The physicochemical and biological properties of a molecule critically depend upon conformations the molecule can adopt. Therefore, carrying out exhaustive and meaningful conformational analysis is pivotal for deeply investigating any molecular feature. For instance, any three-dimensional ligand-based approach in drug design can't help using a complete analysis of the conformational space. Monte Carlo simulation is just one of the methods available to achieve this sampling (Chang, *et al*., 1989). In a Monte Carlo study, the conformational space of a molecule is sampled by randomly changing dihedral angle rotations or atom Cartesian coordinates. If the currently drawn sample is lower in energy than its predecessor, then it is retained as a starting point for the successive iteration. Conversely, when the new conformation is higher in energy, it can be retained according to two alternative criteria: either its energy belongs to a predefined window or the ''move'' can be accepted with a probability related to the Boltzmann factor, following the Metropolis method (Metropolis, *et al*., 1953).

Two fields that make a great use of conformational sampling are docking and virtual screening, both of them holding a prominent position in the modern structure-based drug design (Taylor, *et al*., 2002). In a limited computational time, they have to face a hard two-fold problem: generating a sensible conformational ensemble and then ranking its members. Besides Monte Carlo sampling, the ligand conformational space can be explored by genetic and incremental algorithms.

Apparently, sampling is an easier job to do than scoring. In fact, reiterated docking runs usually provide at least one pose close to the crystallographic one. In contrast, due to different heuristics and approximation levels, scoring functions do not always succeed in including the crystallographic pose among the most favorable ones. On top of it, it is not unusual to see quite different rankings by some among the most widespread docking tools. In general, it cannot be said that one method outperforms the others, since different target and compound classes can lead to different performances. A number of different possibilities rather than a single binding mode can be obtained also as a result of reiterated runs of the same algorithm, when it adopts a random based approach. Due to the computational cost of the sampling process and of the evaluation of the binding free energy, it would be definitely useful to have a restricted, but still representative, set of conformations to be processed with more thorough techniques.

Cluster Analysis (CA) is a discipline that encompasses a number of different algorithms to partition samples in homogeneous classes without any *a priori* knowledge. It is already used to analyze the large amount of data generated by molecular modeling software, such as the outcomes of conformational analysis and docking outputs (Chema, *et al*., 2004).

In principle, there does not exist a unique ''correct'' method to cluster a dataset, and a large number of variations have been devised, from which one has to choose the most appropriate one.

As an example, X-cluster, developed in 1994 by Shenkin and McDonald (Shenkin and McDonald, 1994) and implemented in the MacroModel software package (Mohamadi, *et al*., 1990) is one of the most widely exploited algorithms for organizing the output of conformational sampling. X-cluster employs a hierarchical agglomerative approach with the single linkage rule (see the Algorithm Description section for further details). As a major drawback for any

*To whom correspondence should be addressed.

automated procedure, X-cluster leaves to the user the choice of the most suitable clustering level.

In docking and virtual screening simulations, some programs (such as AutoDock and GOLD) implement CA to better rationalize their outcomes. In particular, AutoDock sorts conformations by increasing energy and then implements a nonhierarchical clustering method with single linkage rule to partition the poses. The clustering process always starts from the best scoring pose, and, due to the peculiarity of the single linkage rule, first clusters tend to be the more numerous. The process is iterated through conformations, grouping together the elements whose Root Mean Square Deviations (RMSDs) are within a user-defined threshold value. In turn, GOLD has a dedicated utility (rms_analysis) to perform CA on the docked poses with a hierarchical agglomerative approach based on the complete linkage rule. This is known to be a non space-conservative linkage criterion that tends to create compact clusters of similar dimension. Moreover, no cutting rule is implemented in the CA of the GOLD program.

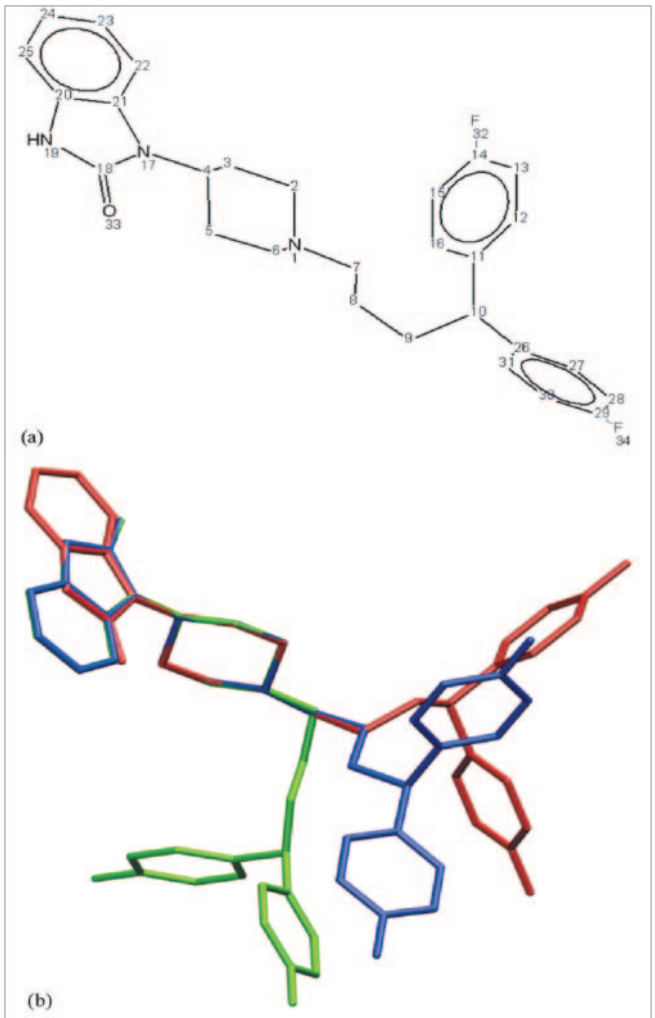
A suitable CA protocol should be able to provide a functional classification, i.e., to identify few conformations worthy to be further studied. Moreover, the protocol should be ''information'' driven and should not, in general, necessitate of any preexisting knowledge about the specificities of the target. Recently, we carried out a comparative study (Bottegoni, *et al*., 2006) about the use of different hierarchical agglomerative clustering rules associated with a user-independent cutting function applied to the outcomes of four different docking programs. From that study, we learned that the combination of an *a priori* clusterability assessment with the average linkage rule, and with a stopping criterion based on the Kelley-Gardner-Sutcliffe (KGS) penalty function (Kelley, *et al*., 1997) provides a good basis to achieve a sensible partitioning of conformational datasets.

In this work, we describe the implementation of our novel protocol in a MATLAB (The MathWorks, Inc.) metalanguage program, named AClAP (Autonomous hierarchical agglomerative Cluster Analysis based Protocol), and we discuss its performance vs. commonly available CA-based methods. AClAP design benefits from the understanding we gained from a conformational analysis we made over a set of ten marketed drugs with the aid of MacroModel (Mohamadi, *et al*., 1990) and over the above mentioned docking results, which concern a quite diverse set of ligands co-crystallized with different biological counterparts. Docking simulations were carried out by means of four programs, namely Dock (Ewing, *et al*., 2001), AutoDock (Morris, *et al*., 1998), GOLD (Jones, *et al*., 1997), and FlexX (Rarey, *et al*., 1996). Moreover, we statistically analyze the whole set of obtained conformations, and finally we discuss the behavior of the KGS penalty function.

Summarizing, AClAP turned out to meet all of the criteria required for a robust clustering protocol at a very limited computational cost. Therefore, we propose it as an innovative and user-friendly tool, which can be of great help to molecular modelers dealing with both ligand- and target-based drug design.

2 METHODS

AClAP is an interactive MATLAB metalanguage program that can take data from the widespread mol2 file format. AClAP can also take in input the torsion angles either in raw or csv (comma separated values) formats. It is able to automatically identify the number of poses and the set of nonhydrogen atoms.



Torsions #	1	2	3	4	5	6	7
Atoms	18-17-4-3	6-1-7-8	1-7-8-9	7-8-9-10	8-9-10-11	9-10-11-12	9-10-26-27
Conf. #1 (blue)	115.93	63.56	58.76	173.89	66.55	-122.36	-54.14
Conf. #2 (green)	116.09	170.48	-58.76	-173.83	166.51	-145.90	-56.72
...							
Conf. #121 (red)	-61.92	68.42	-106.91	-70.16	166.74	-144.22	-55.90

Fig. 1. Example of construction of matrix **M** for Pimozide. **a**) The nonhydrogen atom numbering and the acyclic torsion angles are reported. **b**) Pimozide chemical structure. The parameters reported in the table give rise to *n* by *d* matrix **M**, where *n* is the number of the sampled conformations (in the present example, 121) and *d* is the number of degrees of freedom of the molecule (in the present example, the 7 acyclic torsion angles of Pimozide).

Each conformation is considered as an observation in a *d*-dimensional space, and it is stored in an *n* by *d* matrix **M**, where *n* is the number of the sampled conformations and *d* is the number of degrees of freedom of the molecule (Figure 1). These latter include dihedral angle values of all rotatable bonds and the Cartesian coordinates of three atoms (limited to the clustering of docking poses), which account for global rotation and translation. Each column of **M** is z-standardized for subsequent

```
ACLAP - Version 1.0


"Data, data everywhere...."


Perform H* Test ? [y/n]
Select: y
Are poses superimposed ? [y/n]
Select: y
Poses file: prazosina_fit.mol2

Heavy Atoms Number   28

Parsing poses, please wait....


Poses Detected   24

Torsions file: prazosin.csv
Torsion File Format ? [csv - vector]
Select: csv
More poses to cluster ? [y/n]
Select: n
H* value is :   0.87

Procede to Cluster Analysis ? [y/n]
Select: y
Compare to a Reference Structure ? [y/n]
Select: n
Define symmetric atomic couples ? [y/n]
Select: n

Linkage Method:
[average - single - ward]
Seleziona: average
Plot the KGS Penalty Score Function ? [y/n]
Select: y
Save the current plot ? [y/n]
Select: n

Minimum KGS Penalty Function:     16.9
N. of Clusters:      5
N. of non-singleton Clusters:   5
Most Populated Cluster is n. 2 (10 elements)

Do you want to write a Report output ? [y/n]
Select: y
Report File name: prazosin_rep
```

Fig. 2. The dialog box of AClAP.

processing. The **M** matrix is exploited within the clusterability assessment, whereas, for CA, the full Cartesian coordinate set is used. In Figure 2, a typical dialog box of AClAP is shown.

2.1 Clusterability assessment

To assess whether conformations show a natural tendency to group into clusters, we implemented a modified version of a test originally developed by Hopkins (Hopkins, 1954): the H^* test.

This test is aimed at distinguishing between three main possibilities for the distribution of the members of the dataset: uniformly scattered, regularly spaced or naturally grouping. Only in the last case, CA is really justified. The H^* test is implemented as follows: first, a Principal Component Analysis is performed over the z-standardized matrix **M**; in order to lower the dimensionality of the problem, the original dataset is projected onto the reduced space L induced by the first three principal components. Then, a small number s of random points in L is generated. These points are normally distributed, with zero means, and their projection over each principal component direction has the same standard deviation as the corresponding principal component of the dataset. In our test, $s = n/20$. Now, s poses are randomly drawn and for each of them, as well as for each random point, the minimum distance to the members of the dataset is calculated, and named D_i for the poses, and V_i for the points. This procedure is repeated n times and the H^* value is calculated as the following average:

$$H^* = \left\langle \sum_{i=1}^{s} V_i \Big/ \left(\sum_{i=1}^{s} V_i + \sum_{i=1}^{s} D_i \right) \right\rangle_{dataset}, \qquad (1)$$

Three cases can occur:

$0.5 \leq H^* \leq 0.6$ the poses are homogenously distributed
$H^* \to 0$ the poses are regularly spaced
$H^* \to 1$ the poses show a natural tendency to cluster

A cluster analysis should be carried out only in the last one. The absence of regular or repetitive patterns in the outcomes of conformational analysis and docking simulations makes unlikely the occurrence of the second case.

2.2 Cluster Analysis

AClAP implements a hierarchical agglomerative clustering algorithm. ''Hierarchical'' means that clusters at a higher level are union of clusters at lower levels, while ''agglomerative'' means that clusters never break apart during the formation process. The global hierarchy can be represented by means of a dendrogram, a tree showing different clustering levels, spanning from 1 to n. RMSD is taken as a measure of conformation-to-conformation distance. Therefore, the clustering algorithm starts with n unary clusters; at each step, the two closest clusters are merged, until only one cluster containing all the poses is reached. The way the inter-cluster distance is evaluated is called linkage rule. In AClAP, we implemented three among the most widely used linkage rules: single linkage, average linkage, and the Ward method. Single linkage (Everitt, *et al*., 2001), also known as nearest-neighbor distance method, defines distance as the one of the closest pair of conformations:

$$\Delta_{M,Q} = \min_{m \in \{1,\ldots,\chi_M\}, q \in \{1,\ldots,\chi_Q\}} (d_{m,q}), \qquad (2)$$

where uppercase roman letters indicate clusters, d is the RMSD-based conformation distance, Δ is the inter-cluster distance, χ is the cardinality of a cluster.

A well-known drawback of single linkage rule is the so-called ''chaining'' phenomenon: first clusters naturally tend to incorporate the nearby conformations, therefore forming a ''chain''; as a consequence, there is a strong bias towards the first clusters to being more populated than others.

In the average linkage (Everitt, *et al*., 2001) method, the mean distance between all pairs of conformations is taken:

$$\Delta_{M,Q} = \frac{1}{\chi_M \chi_Q} \sum_{m=1}^{\chi_M} \sum_{q=1}^{\chi_Q} d_{m,q}. \qquad (3)$$

According to this definition no conformation/cluster is preferred with respect to the others, preventing ''chaining'' effect to occur.

Finally, in AClAP, the Ward method can also be selected. This method uses a distance definition based on the analysis of variance (Ward and Hook, 1963). It attempts to minimize the Sum of Squares of any two potential clusters that can be formed at each step. This method tends to create a consistent number of small clusters. Our previous comparative study (Bottegoni, *et al*., 2006) led us to prefer the average linkage rule with respect to both single linkage and the Ward method.

When clustering is finished, the complete dendrogram is obtained and, for each cluster at each level, the so-called centroid can be calculated. The centroid is a ''hypothetical'' conformer whose coordinates are the average coordinates of all the cluster members. The representative conformer for a cluster is chosen as the conformation closest to the centroid. If the homogeneity requirement for the current cluster is fulfilled, the choice of the representative conformer is not expected to be critical.

2.3 Cutting rule

Once the dendrogram is formed, the crucial decision is to fix the level of clustering more suitable to represent the conformational space of interest. As it is natural for a hierarchical agglomerative approach, a tradeoff must be found between the overall number of clusters and the diversity among the conformations that belong to each cluster. AClAP adopts the KGS penalty function. The method is thoroughly described in the paper by Kelley *et al*. (1997) and here summarized and discussed (see Results and Discussion for further details).

An average spread value is calculated for each clustering level of the dendrogram, for simplicity of representation, it is numbered with respect to the number of clusters of the level:

$$AvS_w = \frac{1}{w}\sum_{M=1}^{w} S_M, \tag{4}$$

where w is the number of clusters at a fixed clustering level and S_M is the spread of the M-th cluster, defined as follows:

$$S_M = \frac{2}{\chi_M(\chi_M - 1)}\sum_{m=1}^{\chi_M}\sum_{q=m+1}^{\chi_M} d_{m,q}. \tag{5}$$

When all average spread values are collected, they need to be normalized so that they lie between 1 and $n-1$. The penalty P_w is therefore calculated as:

$$P_w = \frac{(n-2)[AvS_w - \min_{v\in\{1,\ldots,n\}}(AvS_v)]}{Max_{v\in\{1,\ldots,n\}}(AvS_v) - \min_{v\in\{1,\ldots,n\}}(AvS_v)} + w + 1. \tag{6}$$

As expected, this penalty function is a balance between the cardinality of the level and the intra-cluster mean distance. The minimum value of the KGS function can be chosen as an autonomous way (as opposite to a user driven way) to prune the dendrogram. AClAP also provides a detailed description of all local minima occurring before the global minimum is reached; this allows the user to adopt other cutting levels, in the search of more homogeneous clusters.

2.4 Cluster significance

The Chauvenet criterion is often used to determine whether the population of a cluster is statistically significant. According to it, a cluster is significantly populated if its cardinality is more than twice the standard deviation apart from the average population value for that level of clustering. Our rationale for the use of this criterion is to assess whether or not there is evidence that significantly populated clusters deserve particular attention in the conformational analysis and docking contexts.

3 RESULTS AND DISCUSSION

AClAP resulted both in an innovative protocol to autonomously partition conformational datasets, and in a program that accomplishes it in a negligible computational time as compared to that needed to generate the dataset itself. On an Intel PentiumIV (2.4 GHz) processor with 512 MB of RAM, AClAP performed CA over 520 conformations of the drug Fexofenadine in 260 sec. In Figure 3, an example of a typical AClAP report is shown. The program provides the overall number of clusters and how many of them are non singleton. For each cluster, the centroid and the representative conformation, as above defined, are calculated. If a reference conformation, such as a crystallographic pose, is available, AClAP allows a comparison of all the representative poses with it (on an RMSD basis).

3.1 KGS penalty function

With standard options, AClAP uses the global minimum of the penalty function as a cutting criterion of the dendrogram. It is

CLUSTER #	CARD	REPR	CEN.DI	RMSD	CHAU
1	2	219	0.445	5.057	no
2	3	298	0.955	4.782	no
3	45	303	1.763	9.827	yes
4	4	305	2.536	9.922	no
5	32	96	3.538	3.070	no
6	12	97	4.458	3.092	no
7	5	244	4.870	5.134	no
8	11	256	5.678	4.925	no
9	11	274	6.635	4.939	no
10	3	257	6.999	4.632	no

Fig. 3. Excerpt from an AClAP report. Columns refer to: cluster number, cardinality and representative conformation; the distance from this latter to the centroid; the RMSD from a reference conformation and whether the cluster is significantly populated.

interesting to comment about the behavior of this function and about the information it provides. We observe that the function P is the sum of two terms: there is a constant slope term that accounts for the increase of the number of clusters and a term that is proportional to AvS_w. One property that would be of interest for the penalty function is a unique minimum. This property would be guaranteed if the average spreads were monotonically decreasing and concave functions. To this aim, let's consider a single step increment of the average spread, which can be reformulated in the following way:

$$\begin{aligned} AvS_w - AvS_{w-1} = &-\frac{1}{w(w-1)}\sum_{M=1}^{w-2} S_M + \\ &-\frac{1}{w-1}\frac{2}{(\chi_A+\chi_B)(\chi_A+\chi_B-1)}\sum_{a=1}^{\chi_A}\sum_{b=1}^{\chi_B} d_{a,b} + \\ &+\left(\frac{S_A}{w}\right)\left[1-\left(\frac{w}{w-1}\right)\frac{\chi_A(\chi_A-1)}{(\chi_A+\chi_B)(\chi_A+\chi_B-1)}\right] + \\ 0< &+\left(\frac{S_B}{w}\right)\left[1-\left(\frac{w}{w-1}\right)\frac{\chi_B(\chi_B-1)}{(\chi_A+\chi_B)(\chi_A+\chi_B-1)}\right] \end{aligned} \tag{7}$$

>1 <1

Here, we called A and B the clusters merged at the current clustering step. One can see that the increment is given by the sum of four terms, the first one is always negative, but supposedly small, and is related to the average spread of the clusters non involved in the current step. The second one, again negative, is the average spread given by the inter-cluster (A and B) conformations. Third and fourth terms have no fixed sign but it can be assumed that most of the times they are positive. Given the way the clustering algorithm works, monotonicity and concavity would be implied by a second term being always prevalent over the last two. In general, this is not true. But we gain an interpretation clue from this: any time we see a definite decreasing behavior of the penalty function; it

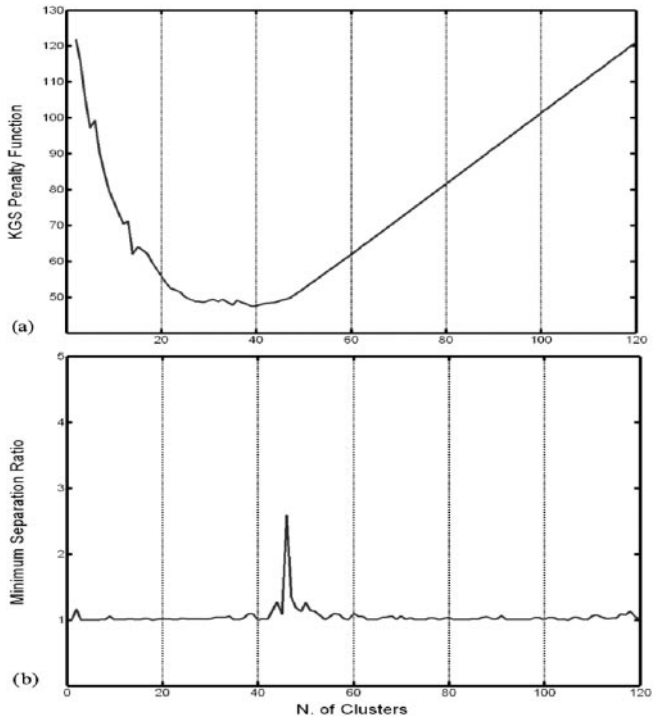


Fig. 4. Cutting rule indicators as implemented in AClAP and X-cluster, they are applied to a member of the ''easy'' drug set: Pimozide (121 conformers). a) The KGS penalty function is plotted vs. the overall cluster number. Standardly, AClAP adopts the minimum of the KGS penalty function as cutting level. b) The MSR value is plotted vs. the overall number of clusters. X-cluster uses the MSR as an indication of different cutting levels. The complete set of clustering results is available as Supplementary Information. These plots show that both algorithms point at a well-defined partitioning.

means that the clustering process has merged two clusters that were well separated. In other words, if one is very much concerned about intra-cluster homogeneity, one has to stop the clustering process at the first pronounced local minimum (which is the rightmost in the plot versus w). Going further on means privileging synthetic representation with respect to intra-cluster homogeneity (see Figures 4a and 6a as typical examples of the KGS behavior).

In what follows, we report the observations we were able to make on the conformational space sampling made both by Metropolis Monte Carlo simulations, and by four docking tools, pointing out how and where they could benefit from the new clustering method.

3.2 Monte Carlo conformational analysis

Metropolis Monte Carlo method (T = 300 K) as implemented in the MacroModel software package was used to perform a conformational analysis on ten marketed drugs. We approximately could split them in conformationally ''easy'' and ''hard'' ones: drugs with up to seven rotatable bonds for which Monte Carlo search provided less than 150 conformers were assigned to the ''easy'' set, whereas the others (with up to 10 rotatable bonds and more than 150 conformers) were defined as ''hard'' compounds.

In the following, we compare the clustering outcomes of AClAP to those of X-cluster (Shenkin and McDonald, 1994),

Table 1. AClAP results for the drug conformations generated via Monte Carlo simulations. For Fentanyl, H^* was 0.53 indicating that CA was not justified. This rule holds more strictly when docking simulations are concerned, whereas drug conformers might still benefit from a CA. Rot. stands for rotatable, NS for NonSingleton and Sign. for significantly populated according to the Chauvenet criterion.

Drug	# confs	Rot. bonds	Max RMSD	H^*	# clusters	NS clusters	Sign. clusters
Conformationally ''easy'' drugs							
Prazosin	24	4	3.24	0.83	5	5	1
Amsacrine	33	5	5.20	0.67	12	11	0
Citalopram	37	4	2.80	0.80	19	14	2
Mizolastine	47	5	4.10	0.65	14	13	0
Fentanyl	49	7	4.26	0.53	12	9	0
Pimozide	121	7	7.30	0.62	39	29	1
Conformationally ''hard'' drugs							
Astemizole	235	8	6.44	0.63	24	24	0
Bepridil	285	9	5.90	0.80	40	35	2
Dofetilide	414	10	10.86	0.77	44	30	3
Fexofenadine	520	9	8.59	0.84	74	58	4
Astemizole	235	8	6.44	0.63	24	24	0

a commonly used clustering procedure implemented in the MacroModel software package. X-cluster is a hierarchical agglomerative clustering method that adopts the single linkage rule. It provides the user with the Minimum Separation Ratio (MSR), which is a function aimed at suggesting a clustering level where all the clusters are well separated. If the MSR is less than 1, the partitioning is expected to be poor. In contrast, an MSR value greater or equal to 2 is an indication of a good partitioning. The final choice of the clustering level is however left to the user.

Preliminarily to our comparison, we adopted the Corrected Rand Index (Hubert and Arabie, 1985) in order to evaluate the similarity of their results. This index is a common measure of the difference between partitionings of the same data set, and it ranges between 0, indicating a strong divergence, and 1, indicating partitioning coincidence.

For the conformationally ''easy'' drugs of the series, Prazosin, Amsacrine, Citalopram, Mizolastine, Fentanyl, and Pimozide, AClAP was able to indicate a functional partitioning, while X-cluster had success in 5 out of 6 cases. AClAP decided for the best clustering level according to the minimum of the KGS penalty function. In Table 1, overall results of AClAP are reported, while Figure 4 shows the AClAP (Figure 4a) and X-cluster (Figure 4b) outcomes applied to the 121 conformers of Pimozide, taken as a representative example for the set of ''easy'' drugs. Figure 4b clearly indicates that, in the reported example, MSR was able to point to a plausible partitioning.

Partitioning obtained by X-cluster applied to conformationally ''easy'' drugs is summarized in Table 2. In particular, for Prazosin, Citalopram, and Pimozide, the MSR values pointed univocally to a cutting level for the hierarchical tree. The partitionings strongly agree with those obtained by AClAP, the Corrected Rand Index values being 0.74, 0.79, and 0.79, for the three molecules, respectively (see the last column of Table 2). Conversely, for Fentanyl, the MSR value provided no clear indication of a cutting level. The H^* value for Fentanyl provided by AClAP was 0.53 (see

Table 2. X-cluster results for 5 ''easy'' drugs, whose conformations were generated via Monte Carlo simulations. X-cluster did not provide any significant cutting point for Fentanyl.

Drug	# conformers	MSR	# clusters	NS Cluster	Corrected Rand Index
Prazosin	24	1.90	4	4	0.74
Amsacrine	33	4.42	2	2	0.03
Citalopram	37	19.40	21	16	0.79
Mizolastine 1	47	2.00	2	2	0.15
Mizolastine 2	47	1.93	18	14	0.90
Pimozide	121	2.58	47	33	0.79

Table 1), suggesting that conformations did not display a natural tendency to aggregate into groups. It should be mentioned that, when H^* is less than 0.6, unlike structure-based drug design, ligand-based drug design might still benefit from CA applied to drug conformers. Consistently, AClAP applied on Fentanyl provided a quite good partitioning, as reported in Table 1. In the case of another ''easy'' drug, Amsacrine, a significant MSR value led to a partition with only two clusters. Conversely, the partition provided by AClAP afforded 12 clusters. The Corrected Rand Index was as low as 0.03, indicating that the two partitionings were markedly different (see Figures 5a and 5b). As it can be seen in Figure 5b, the internal homogeneity of the partitioning provided by X-cluster was rather poor. One possible reason could be the chaining effect induced by the single linkage rule. Finally, in the analysis of Mizolastine, two clustering levels worthy to be selected were identified, showing MSR values of 2 and 1.93, respectively (Mizolastine 1 and Mizolastine 2 in Table 2). The clustering of Mizolastine 1 (MSR = 2) corresponded to a partition with only two clusters, lacking internal homogeneity and displaying an evident chaining effect (data not shown). The second partitioning (Mizolastine 2, 18 clusters, 14 nonsingletons) provided more homogenous clusters and a strong agreement with the partition obtained by AClAP (Corrected Rand Index = 0.90).

When processing the conformationally ''hard'' drug set (composed by Astemizole, Bepridil, Dofetilide, and Fexofenadine), whose conformers were generated via Metropolis Monte Carlo simulations, X-cluster did not provide any clue about the cutting level for the conformations, demonstrating that a protocol based on the single linkage rule in combination with MSR fails when dealing with conformationally complex molecules. In Figure 6, as an example, the 520 conformers of Fexofenadine treated with AClAP (Figure 6a) and X-cluster (Figure 6b) are shown. As reported in Table 1, in these cases H^* test showed a natural grouping tendency, and AClAP, a protocol based on the average linkage rule in combination with the KGS penalty function was actually able to univocally provide a good partitioning for all the drug conformers (see Figure 6a and Table 1). We can conclude that, for the drugs here investigated, AClAP definitely outperformed X-cluster.

3.3 Docking simulations

We studied the conformational sampling done by four among the most widespread docking programs, namely, Dock, AutoDock,

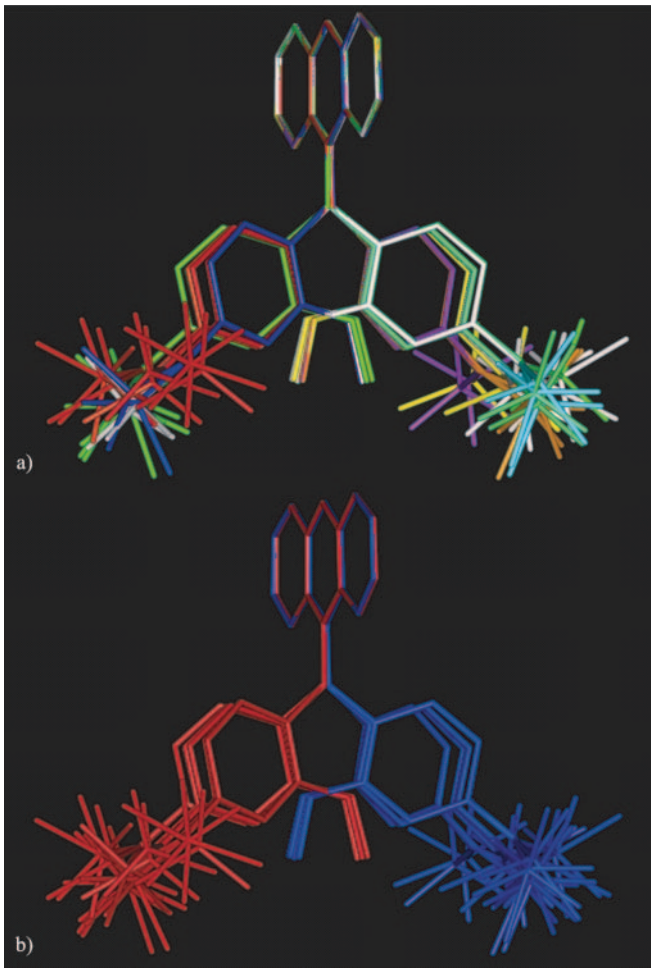


Fig. 5. The partitioning of 33 conformers of Amsacrine. **a**) The partitioning provided by AClAP. A protocol based on the average linkage rule and the KGS cutting function generated 12 groups bearing high intra-cluster homogeneity. **b**) The partitioning provided by X-cluster. A protocol based on the single linkage rule and a user-dependent cutting function generated 2 groups bearing scarce intra-cluster homogeneity.

GOLD, and FlexX, together with the action of our clustering protocol over their output. We ran the programs over a set of 16 crystallographic complexes belonging to the following protein families: kinases, hormone receptors, and proteases (both serine and aspartic proteases). As a figure of merit, we took the RMSD of the generated poses from the crystallographic one. For a detailed description of docking simulations and comparative analysis the reader is referred to the work of Bottegoni *et al*. (Bottegoni, *et al*., 2006). In what follows, we summarize some conclusions we drew from that experience. The present comments encompass only 15 cases, since one of the original ones (Propidium co-crystallized with AChE, PDB code 1N5R) has been demonstrated to bind to the surface of its biological counterpart in at least two different modes (Bourne, *et al*., 2003; Cavalli, *et al*., 2004).

About conformational sampling, and having defined a ''good'' pose as the one which is less than 2.5 Å far away (in terms of RMSD

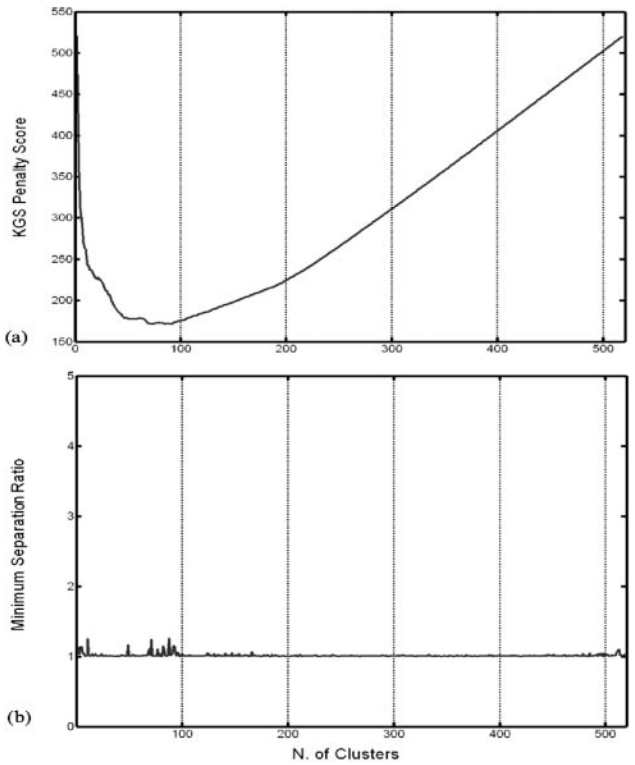


Fig. 6. The cutting rule as implemented in AClAP and X-cluster. As an example of complex, ''hard'', compound, the CA was performed over 520 conformers of Fexofenadine. **a**) The KGS penalty function is plotted vs. the overall number of clusters. Standardly, AClAP adopts the minimum of the KGS penalty function as cutting level. **b**) The MSR value is plotted vs. the overall number of clusters. X-cluster uses the MSR as an indication of possible cutting levels. These plots show that only the KGS score was able to point at a well-defined partitioning. The complete set of clustering results is available as Supplementary Information.

between nonhydrogen atoms) from the crystallographic one, we can comment as follows:

- at least one among the docking tools was always able to generate a pose sufficiently close to the experimental one, being of 89% the average success rate among docking programs; in particular, AutoDock and GOLD were always able to provide at least one ''good'' pose, whereas Dock and FlexX had a success rate of 80% and 73%, respectively;
- however, in terms of conformational sampling, no docking tool significantly outperformed the others, with a chi-square value of 0.67, corresponding to a 88% of probability that fluctuations in the results are random.

Comments about clustering features, shown in more detail in Table 3, follow:

- the best pose was found within a singleton cluster with a very low frequency, ranging from 0.2% to 0.7%;
- when a single docking tool was used, the assertion that ''good'' poses are to be found only, or mainly, in the most populated clusters did not find any clear evidence;
- when a ''holistic'' approach was adopted, i.e., the clustering was performed over the poses generated by all of the docking tools, the probability of finding at least one ''good'' pose among the representative conformations of the most populated clusters, whose number was always between 1 and 3, reached roughly 93%;
- in the holistic approach, as compared with the single tool approach, the presence of ''good'' poses decreases in scarcely populated clusters in favor of very highly populated ones.

A comment is due about the performance of AClAP in the so-called holistic approach. No scoring process was used to support the provided results. Nevertheless, their performance can be compared to that of the widely used consensus scoring method, which well overcomes the main limitation of scoring functions. Indeed, also in our investigation, the scoring functions sometimes failed to rank correctly the best poses: roughly in the 50% of the cases. We found of particular interest the data shown in the last two columns of Table 3: they indicate that, at least for the molecules we examined, there is a high chance to find a ''good'' pose among the representative conformations of the most populated clusters. According to our arrangement procedure in bins, and similarly to what obtained with the Chauvenet criterion, those conformations are usually less than two. This procedure seems to point at a few, but still very promising, candidates that can be successively examined with more accurate tools, providing a really remarkable dimensionality reduction.

4 CONCLUSIONS

In this paper, we have described a new clustering protocol as well as its implementation in a MATLAB program. The new software, named AClAP, turned out to be well suited to cluster both conformations generated via Metropolis Monte Carlo simulations of drugs, and poses obtained by reiterated docking runs. In a consistent fashion, AClAP prompts the user to assess the clusterability of a conformational dataset by means of what we named the H^* test. The subsequent step is a hierarchical agglomerative cluster analysis based on the average linkage rule. The choice of this rule with respect to others was already discussed elsewhere (Bottegoni, *et al*., 2006), and here reinforced. Once the hierarchical tree is built, an autonomous method to prune it is needed to define the best clustering level. Here, we have shown that the KGS penalty function is an unbiased approach very well suited to achieve that goal. AClAP outperforms standard CA-based protocols as they are implemented in the most commonly used docking programs. In this context, the AClAP method manages to greatly reduce conformational space dimensionality, proving to be fruitful, for instance, for the successive application of computationally intensive energy estimation techniques to be applied to cluster representatives. On top of it, in what we called the holistic approach, AClAP allowed us to identify some one among the closest poses to the experimental one, and placed it within a statistically significant cluster with a very promising hit rate. Finally, when applied to the output of Metropolis Monte Carlo searches, AClAP proved to be more robust than the long-time exploited and commonly used X-cluster routine. Encouraged by the present results, we propose AClAP as a new and user-friendly tool to help molecular modelers facing issues related to both ligand- and target-based drug design. Our efforts are currently devoted to extend the appli-

Table 3. Distribution of ''good'' poses with respect to relative cluster cardinality. For each docked molecule, at the clustering level chosen by AClAP, the clusters are classified in five bins (A to E) according to their cardinality, in ascending order. Then, for each molecule, the relative frequency of the ''good'' poses, i.e., those with an RMSD < 2.5 Å from the crystallographic one, is calculated. The derived frequencies, at fixed docking program, are then averaged over the different molecules. In the last row, the outcome of the holistic approach is reported. In parentheses, the bin population with respect to the total cluster number is shown. As one can see, the ''good'' poses tend to be distributed among very highly and very scarcely populated clusters, with a prevalence of the formers. The holistic approach seems to make this prevalence maximally marked.

	Class A (least populated clusters) %	Class B %	Class C %	Class D %	Class E (most populated clusters) %	Good poses in singleton clusters %	Frequency of at least one ''good'' pose in a signif. populated cluster %.	Frequency that at least one representative pose of a cluster in E bin is ''good'' %	Average number of clusters in E bin
AutoDock	12.6 (86.6)	9.0 (5.7)	6.4 (1.6)	7.6 (1.5)	64.4 (4.6)	0.3	80.0	80.0	1.1
FlexX *	33.0 (72.0)	23.8 (14.1)	5.5 (6.2)	0.8 (2.2)	36.9 (5.5)	0.2	33.3	40.0	1.3
Dock *	31.2 (83.4)	10.8 (4.4)	5.9 (1.1)	2.2 (1.0)	49.9 (10.1)	0.7	53.3	50.0	1.1
GOLD	14.9 (76.1)	7.3 (7.1)	18.3 (5.0)	1.8 (0.2)	57.7 (11.6)	0.5	40.0	78.6	1.2
Average	**22.9 (79.5)**	**12.7 (7.8)**	**9.0 (3.5)**	**3.1 (1.2)**	**52.2 (8.0)**	**0.4**	**51.7**	**62.1**	**1.2**
Holistic	**14.3 (94.3)**	**6.8 (2.3)**	**11.1 (1.2)**	**2.3 (0.4)**	**65.5 (1.8)**	**0.2**	**100.0**	**93.3**	**1.1**

*In one case, these programs weren't able to find any pose closer than 2.5 Å to the crystallographic one.

cability of this approach to rationalize the outcomes of protein-protein docking.

ACKNOWLEDGEMENTS

We thank M. Masetti for technical assistance. Miur-Cofin2004 and FIRB projects (''Sviluppo di metodologie innovative per l'identificazione e la sintesi di nuove molecole a scopo terapeutico: applicazioni nel campo della malattia di Alzheimer'' and ''Laboratorio Nazionale sulle Nanotecnologie per Genomica e Postgenomica (NG-Lab)'') are gratefully acknowledged for the financial support.

REFERENCES

Bottegoni,G., Cavalli,A. and Recanatini,M. (2006) A comparative study on the application of hierarchical-agglomerative clustering approaches to organize outputs of reiterated docking runs. *J. Chem. Inf. Mod.*, **46**, 852–862.

Bourne,Y., Taylor,P., Radic,Z. and Marchot,P. (2003) Structural insights into ligand interactions at the acetylcholinesterase peripheral anionic site. *Embo J.*, **22**, 1–12.

Cavalli,A., Bottegoni,G., Raco,C., De Vivo,M. and Recanatini,M. (2004) A computational study of the binding of propidium to the peripheral anionic site of human acetylcholinesterase. *J. Med. Chem.*, **47**, 3991–3999.

Chang,G., Guida,W.C. and Still,W.C. (1989) An internal coordinate Monte Carlo method for searching conformational space. *J. Am. Chem. Soc.*, **111**, 4379–4386.

Chema,D., Eren,D., Yayon,A., Goldblum,A. and Zaliani,A. (2004) Identifying the binding mode of a molecular scaffold. *J. Comput. Aided Mol. Des.*, **18**, 23–40.

Everitt,B.S., Landau,S. and Leese,M. (2001) Cluster Analysis. Arnold, a member of the Hodder Headline Group,London.

Ewing,T.J., Makino,S., Skillman,A.G. and Kuntz,I.D. (2001) DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases. *J. Comput. Aided Mol. Des.*, **15**, 411–428.

Hopkins,B. (1954) A new method for determining the type of distribution of plant individuals. *Ann. Bot.-London*, **18**, 213–227.

Hubert,L.J. and Arabie,P. (1985) Comparing partitions. *J. Classif.*, **2**, 193–218.

Jones,G., Willett,P., Glen,R.C., Leach,A.R. and Taylor,R. (1997) Development and validation of a genetic algorithm for flexible docking. *J. Mol. Biol.*, **267**, 727–748.

Kelley,L.A., Gardner,S.P. and Sutcliffe,M.J. (1997) An automated approach for defining core atoms and domains in an ensemble of NMR-derived protein structures. *Protein Eng.*, **10**, 737–741.

Metropolis,N., Rosenbluth,A.W., Rosenbluth,M.N., Teller,A.H. and Teller,E. (1953) Equation of state calculations by fast computing machines. *J. Chem. Phys.*, **21**, 1087–1092.

Mohamadi,F., Richards,N.G.J., Guida,W.C., Liskamp,R.M.J., Lipton,M.A., Caulfield,C.E., Chang,G., Hendrickson,T.F. and Still,W.C. (1990) MacroModel—an integrated software system for modeling organic and bioorganic molecules using molecular mechanics. *J. Comput. Chem.*, **1**, 440–467.

Morris,G.M., Goodsell,D.S., Halliday,R.S., Huey,R., Hart,W.E., Belew,R.K. and Olson,A.J. (1998) Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.*, **19**, 1639–1662.

Rarey,M., Kramer,B., Lengauer,T. and Klebe,G. (1996) A fast flexible docking method using an incremental construction algorithm. *J. Mol. Biol.*, **261**, 470–489.

Shenkin,P.S. and McDonald,D.Q. (1994) Cluster analysis of molecular conformations. *J. Comput. Chem.*, **15**, 899–916.

Taylor,R.D., Jewsbury,P.J. and Essex,J.W. (2002) A review of protein-small molecule docking methods. *J. Comput. Aided Mol. Des.*, **16**, 151–166.

Ward,J.H.J. and Hook,M.E. (1963) Application of a hierarchical grouping procedure to problem of grouping profiles. *Educ. Psychol. Meas.*, **23**, 69–92.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e66–e73
doi:10.1093/bioinformatics/btl266

A top-level ontology of functions and its application in the Open Biomedical Ontologies

Patryk Burek[1,3,†], Robert Hoehndorf[2,3,†], Frank Loebe[1,3,†], Johann Visagie[2], Heinrich Herre[1,3] and Janet Kelso[2,*]

[1]Department of Computer Science, Faculty of Mathematics and Computer Science, University of Leipzig, Augustusplatz 10–11, 04109 Leipzig, [2]Department of Evolutionary Genetics, Max Planck Institute for Evolutionary Anthropology, Deutscher Platz 6, 04103 Leipzig and [3]Research Group Ontologies in Medicine (Onto-Med), Institute of Medical Informatics, Statistics and Epidemiology (IMISE), University of Leipzig, Härtelstraße 16–18, 04107 Leipzig

ABSTRACT

Motivation: A clear understanding of functions in biology is a key component in accurate modelling of molecular, cellular and organismal biology. Using the existing biomedical ontologies it has been impossible to capture the complexity of the community's knowledge about biological functions.

Results: We present here a top-level ontological framework for representing knowledge about biological functions. This framework lends greater accuracy, power and expressiveness to biomedical ontologies by providing a means to capture existing functional knowledge in a more formal manner. An initial major application of the ontology of functions is the provision of a principled way in which to curate functional knowledge and annotations in biomedical ontologies. Further potential applications include the facilitation of ontology interoperability and automated reasoning. A major advantage of the proposed implementation is that it is an extension to existing biomedical ontologies, and can be applied without substantial changes to these domain ontologies.

Availability: The Ontology of Functions (OF) can be downloaded in OWL format from http://onto.eva.mpg.de/. Additionally, a UML profile and supplementary information and guides for using the OF can be accessed from the same website.

Contact: bioonto@lists.informatik.uni-leipzig.de

1 INTRODUCTION

Ontologies play an increasingly important role in modern biology. Recent years have seen a significant expansion in the number of biomedical ontologies and controlled vocabularies. The Open Biomedical Ontologies (OBO)[1] project serves as an umbrella organization providing some basic criteria and guidelines for the standardization of biomedical ontologies.

The OBO project includes a large number of domain-specific ontologies such as the Gene Ontology (GO) (Ashburner *et al*., 2000)—which provides information about processes, molecular functions and sub-cellular locations of genes and gene products—and anatomical and developmental ontologies available for specific species.

Recently, several methodological approaches were discussed which aim to provide an ontological foundation for medical and biomedical domains by means of top-level ontologies (Heller and Herre, 2004b; Smith *et al*., 2005). A top-level ontology explicitly provides domain-independent notions. According to the principles of ontological foundation as expounded in (Heller *et al*., 2004; Heller and Herre, 2004b) and applied in (Herre and Heller, 2005), we pursue the idea of adding top-level layers to existing biomedical ontologies. These layers analyze and formalize general aspects of concepts occurring in these ontologies. The use of a top-level ontology potentially leads to fewer errors in the curation and creation of domain ontologies, a better understanding of the biological concepts and the means for data and ontology integration.

A number of top-level concepts used frequently in various OBO ontologies remain unanalyzed and undefined. Concepts like ''role'' (such as ''oxygen accumulator'') or ''function'' (such as ''to accumulate oxygen'') serve as examples of unanalyzed top-level categories in the OBO ontologies.

Nevertheless, the notion of function is widely used in biomedical ontologies. Most commonly, one of the three hierarchies in the GO is the molecular functions taxonomy. Although the GO provides a short definition for its notion of molecular function, an in-depth analysis is not provided. Further uses of the notion of function appear in the Chemical Entities of Biological Interest (ChEBI) Ontology (Brooksbank *et al*., 2005) and in the Celltype (CL) Ontology (Bard *et al*., 2005), equally without a strong theoretical basis concerning functions.

We believe that a theory of functions is useful for the development and application of biomedical ontologies. To date, criticisms of the use of the concept of function in biomedical ontologies either proposed no solution or implied extensive changes, or a complete refactoring of existing structures (Smith *et al*., 2003). Considering the GO's molecular function taxonomy, for example, we realize that this poses problems for a resource under constant usage by the community. Therefore we propose to address this problem in another way.

*To whom correspondence should be addressed.
†These authors contributed equally to this work.
[1]http://obo.sourceforge.net

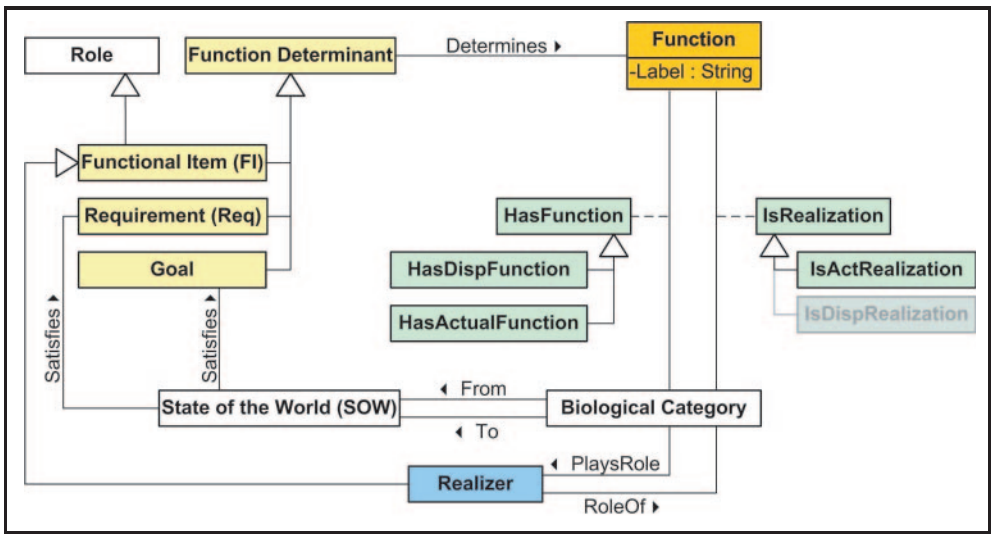


Fig. 1. A schematic representation of the concepts used and introduced by the OF (using the Unified Modeling Language (OMG, 2006)). Unlabelled relations indicate generalizations, where large arrowheads point at the more general concept. Functions (the orange box) are determined by entities indicated in yellow: a goal, requirements, and a functional item. A biological category may be related to a function in two ways (cf. the green boxes which provide labels for those relations connected to them by a dashed line): its instances may *realize* the function or they may *have* the function. A biological entity (such as a process) is a realization of a function if it mediates between two states of the world, one satisfying the requirements, the other satisfying the goal. A realizer in the OF, presented in blue, is the role played by an entity in a realization. In the function this role is determined by the functional item, hence realizer is generalized by functional item. Biological categories whose instances can play the role defined by the functional item have the function. The *HasFunction* relation relates biological categories with functions if every instance of this category has the actual or dispositional function.

We describe here a proposal for a top-level ontology of biological functions. This proposal introduces functions as an additional layer to the existing biomedical ontologies. We consider this ontology orthogonal to those currently in use. Although concepts relating to functions exist in biomedical ontologies, they are not yet adequately presented in an explicit, formal manner. Using our framework, this missing knowledge can be introduced in the existing biomedical ontologies while preserving their original structure.

For this purpose we introduce new relations such as *Has Function* and *IsRealization* in order to relate concepts of existing biomedical ontologies to functions as modelled in our approach. These relations and the specification of the structure of functions capture, in a separate ontology, information which is present at the stage of annotation.

We demonstrate the application of the proposed Ontology of Functions by showing how it can be used to systematically add explicit links between molecular functions and biological processes in the GO. We will further apply the formalism to the Celltype Ontology (CL), and will show how our proposal can serve to make definitions in CL precise, identify entities which are not yet covered by CL, and thereby contribute to CL's completeness.

Finally, we discuss advantages of our approach. In particular, we focus on the extent to which it may aid automated reasoning and data integration.

2 RESULTS

2.1 Introduction to the Ontology of Functions

We introduce here selected concepts of the *Ontology of Functions* (OF), which are presented in detail in (Burek, 2006). The OF will be included as a module in the *General Formal Ontology* (GFO; cf.(Heller and Herre, 2004a; Herre *et al*., 2006)), a top-level ontology developed and maintained by the research group Ontologies in Medicine (Onto-Med)[2]. The OF aims to provide a domain-independent, conceptual framework for the representation of knowledge about functions. An overview of the main concepts and relations introduced by the OF is given in Figure 1.

In an adaption of (Searle, 1995; Sasajima *et al*., 1995), we consider functions as the abstraction of biological processes or other entities towards a goal: when X has the function Y with the goal Z, then X is supposed to cause or otherwise bring about the state of the world Z, thus realizing Y.

For example, it may be the case that a red blood cell transports oxygen. But the statement that ''the function of the red blood cell is to transport oxygen'' adds a goal or purpose to this description: the red blood cell is *supposed* to transport oxygen – even if the red blood cell is in a condition where it cannot perform this transport, i.e., it is malfunctioning.

The OF addresses three major issues concerning functions:

(1) *function structure*: how to represent and determine functions independently of their realizations

(2) *realization*: the conditions under which a given entity realizes a function

(3) *has-function relation*: the determination of the notion of an entity having a function

Two main assumptions underlie the OF: the separation of functional knowledge from non-functional and the top-level orientation.

[2]http://www.onto-med.de

Concerning the first, in the literature functional knowledge is often considered as providing information about *what* an entity does or what goal it serves, whereas non-functional knowledge describes the structure or behavior of entities and thus answers the question *how* an entity behaves, exists, or realizes functions (Iwasaki and Chandrasekaran, 1992; Rosenman and Gero, 1998). We consider these kinds of knowledge as highly independent, i.e., a function can be described independently of its realization and vice versa. Regarding the second aspect, we view the notions of function, realization, and the has-function relation as common to various domains. The OF therefore qualifies as a top-level ontology rather than a domain ontology.

These two aspects impact further on the application of the Ontology of Functions. The separation of functional and non-functional knowledge permits the application of the OF to domain ontologies without significant changes to them. The top-level orientation, on the other hand, allows for the reuse of the OF across various domains.

2.2 The structure of functions

The OF provides a formal way to represent functions independently of their realizations. The corresponding representation scheme is called a *function structure*. It consists of a set of *labels*, a set of *requirements*, a *goal*, and a *functional item*. Except for the labels, these form the *function determinants*.

Labels are natural language expressions which name the function. Most commonly, phrases of the form ''to do something'' serve as labels, e.g. ''to transport oxygen''.

The *requirements* of the function contain all the necessary preconditions which must be met whenever the function is to be realized. For example, in case of the function ''to transport oxygen from A to B'', oxygen must exist at location A.

Functions are *goal*-oriented entities—specifying a function requires providing the goal it serves. The goal of the function is a state or part of the world—temporally extended or not—which is intended to be achieved by any realization of the function. In the case of transporting oxygen, the location of the oxygen at B is the goal.

The goal specifies only the part of the world directly affected by the function. Often a goal is embedded in a wider context, called *final state*. A final state of a function contains the goal plus an environment for the goal, therefore making the goal more comprehensible.

Functions are dependent entities, in the sense that a function is always the function of some other entity. The *functional item* is a role played by this entity in any realization of the function. In the case of ''to transport oxygen'', it would be an oxygen transporter. The notion of roles is required to explain the nature of a functional item more comprehensively.

We adopt the theory of roles developed in (Loebe, 2003, 2005) and incorporated into the GFO. Accordingly, roles are entities played by a *role player* in a *role context*. For example, ''oxygen transporter'' refers to a role in the role context of ''oxygen transport'', and this role may be played by a red blood cell. This example can further be used to illustrate the dependence relationships of roles. Generally, roles and their role contexts are mutually dependent, i.e., one cannot exist without the other. In contrast, the dependence of roles on their players is one-sided because the player could exist without playing a particular role. In our example, an oxygen transport necessarily involves a oxygen transporter and vice versa. A red blood cell may or may not transport oxygen, thus be playing or not playing the role of oxygen transporter. If it does not play that role, the cell still remains a red blood cell due to other properties such as its histology.

In OF, functional items are special roles which appear in the realization of functions. Note that usually there are more roles involved in the realization of a function than a single role given by the functional item. In a transport process, for example, in addition to the role ''transporter'' there is a ''cargo'' role, referring to that which is transported. Hence, the functional item singles out a particular role whose player is the entity *having* the function.

2.3 Realization and realizer

After introducing the structure of functions, their realization forms the second issue addressed in OF. The notion of realization refers to the question of how the goal of the function is to be accomplished. The *realization* is an entity which provides a transition from the state of the world in which the requirements of the function are fulfilled, to the state in which the goal of the function is fulfilled. This will usually be a process such as an—observed or measured—oxygen transport, but could be another kind of entity such as an instantaneous change.[3]

It is important to understand the difference between a function and a realization, in particular regarding their specification: to specify a function and its structure one has to state *what* is to be achieved; representing a realization usually means to specify *how* something is achieved.

Apart from individuals, it is even more relevant for biomedical ontologies to relate categories directly, such as the process category ''transport'' to the function ''to transport''. The relation *IsRealization* is introduced for this purpose. If a process category stands in the *IsRealization* relation to some function, then all instances of this category are realizations of the function. For example, the category ''oxygen transport'' (a process) stands in the *IsRealization* relation to the function ''to transport oxygen''.

Next, there is a counterpart of functional items on the level of realizations. A functional item is defined as a special role in every realization of a function. It is, in every case, a category (similarly to roles as discussed in (Guarino and Welty, 2000)). In the example of ''to transport oxygen'', the role ''oxygen transporter'' is the functional item. Now consider an individual transport process, i.e., a realization, involving a single red blood cell. That cell has the role ''oxygen transporter'' within this realization. This fact gives rise to a new entity which mediates between the realization and the cell itself, namely the cell as an ''oxygen transporter'' (cell-qua-oxygen transporter). Such an entity is called the *realizer* of the function. Moving to the terminology of roles, we consider realizers to be qua-individuals, i.e., instances of roles (Masolo *et al*., 2005, 2004; Loebe, 2005).

[3]The full framework of OF distinguishes two types of realizations, actual and dispositional. Realizations as introduced here would be called ''actual'' in OF. Dispositional realizations are structurally similar to actual realizations in that they instantiate the same category. For a full discussion, see (Burek, 2006).

In summary, a realization corresponds to a function as a whole, whereas a realizer corresponds to the functional item of that function. The realizer is a qua-individual played by the entity which *has the function*. This leads us to the third major concept of the OF, the *HasFunction* relation.

2.4 Has-function

We address here the question under which conditions a function can be ascribed to an entity. In order to represent function ascription, a ternary relation *has-function* is introduced. This relation takes an individual, a function and a context as arguments. The connection between the first two arguments is such that the individual is involved in a realization of the function as the realizer (e.g., the red blood cell in an oxygen transport process realizing ''to transport oxygen''.)

The context argument reflects the intuition that a function is always ascribed in some context. That means, an individual does not necessarily have a given function in all contexts. For example, a hammer on a pile of papers on a desk may have the function of holding paper, while in the context involving a nail and a wall the function is different. It is out of scope of this paper to investigate the nature of contexts (McCarthy and Buvać, 1998; Akman and Surav, 1996) and we will not include it in this proposal but rather use the has-function relation as if it were a binary relation. However, the background theory surrounding the OF (Burek, 2006) allows for the use of a context argument in the function ascription.

The has-function relation appears in two versions: *actual has-function* and *dispositional has-function*. An entity has an actual function, if it is the role player of the realizer in a realization of the function. If an individual red blood cell is currently transporting oxygen, it has an actual function. If that red blood cell is not transporting oxygen, yet is structurally similar to red blood cells which have that function (by means of being an instance of the same category ''red blood cell''), the non-transporting blood cell is said to have the dispositional function ''to transport oxygen''.

Further, a relation between categories is derived from the has-function relation. A category stands in the *HasFunction* relation to a function, if every instance of the category has that function, actually or dispositionally. For example, ''red blood cell'' is in the *HasFunction* relation to the function ''to transport oxygen''.

Having dealt with the three major issues in the OF—function (structure), realization, and function bearers—let us briefly return to the notion of a realizer, which is considered as a qua-individual. Entities of this kind are not present in the current biomedical ontologies, but they are required in order to link entities which can have functions to realizations. In order to remain consistent with already existing categories of biomedical ontologies we introduce a ternary relation among categories. *Realizes(E, F, R)* represents the fact that entities of the category *E* can play the role of the realizer of the function *F* in realizations of type *R*. For instance, *Realizes*("red blood cell", ''to transport oxygen'', ''oxygen transport'') means that, intuitively speaking, red blood cells can realize the function ''to transport oxygen'' in an ''oxygen transport'' process.

The introduction of a ternary relation—*Realizes*—offers the highest degree of coherence and precision. *Realizes(E,F,R)* entails *IsRealization(R, F)* as well as *HasF unction(E, F)*, while one cannot conclude *Realizes(E,F,R)* from *IsRealization(R, F)*, *HasFunction (E,F)*, and the fact that *E* can participate in *R*. To see why this is the case, consider the general function ''to transport'' (*F*). Red blood cells (*E*) can be said to have this function if we think of an oxygen transport. However, consider a process in which red blood cells are transported, e.g. in the context of some experiment. This transport *R* is a realization of the function and red blood cells are involved in it. However, here they play the role of the cargo rather than that of the transporter. Accordingly, *Realizes(E, F, R)* does not hold in this context and cannot be inferred, even given all other facts.

2.5 Relations between functions

Based on the framework developed in (Burek, 2006) we can introduce relations between functions. Some of the relations introduced are common ontological relations such as subsumption, instantiation, or the part-of relation. For example, the subsumption of functions is defined in terms of the subsumption between the appropriate function determinants.

We can also define new relations which are characteristic only for functions:

- *Support* – one function supports the other if its goal fulfills *partially* the second function's requirements (the goal of the first function is a proper part of the requirements of the second function).
- *Enable* – one function enables the other if its goal fulfills *all* of the second function's requirements (the requirements of the second function are a part of the goal of the first function).
- *Prevent* – one function prevents the other if its goal excludes the requirements of the second.

In (Burek, 2006), more relations between functions are defined, which affect the realizations of functions. For example, one function may *trigger* or *improve* the realization of other functions.

2.6 Application to OBO's ontologies

We explore here potential applications of the Ontology of Functions, and investigate when and where it may be beneficial to use its framework.

2.6.1 Identifying links between processes and functions Our first application is the identification and explanation of relations between processes and functions. The Gene Ontology (Ashburner *et al*., 2000) provides a prime example in this respect. There has been some controversy and discussion about whether the ''Molecular Function'' taxonomy of the Gene Ontology describes functions or activities, and how functions are related to processes (Smith *et al*., 2003). To our knowledge, no practical or theoretical solution has yet been proposed. Functions and activities are usually considered different entities, and actions or activities may realize certain functions. Therefore, while the function of an enzyme may be ''to catalyze'' a reaction, the activity performed by the enzyme is the catalysis itself, which may be embedded in another process.

We assume that at least parts of the Molecular Function taxonomy refer to genuine functions in the sense of the OF, and the annotation relation for some of the gene products annotated to these terms corresponds to the *HasFunction* relation.

A general example is GO:0005215 (transporter activity), which we understand as referring to the function ''to transport''. A more specific example is GO:0051119 (sugar transporter activity), which can be understood as the function ''to transport sugar''.

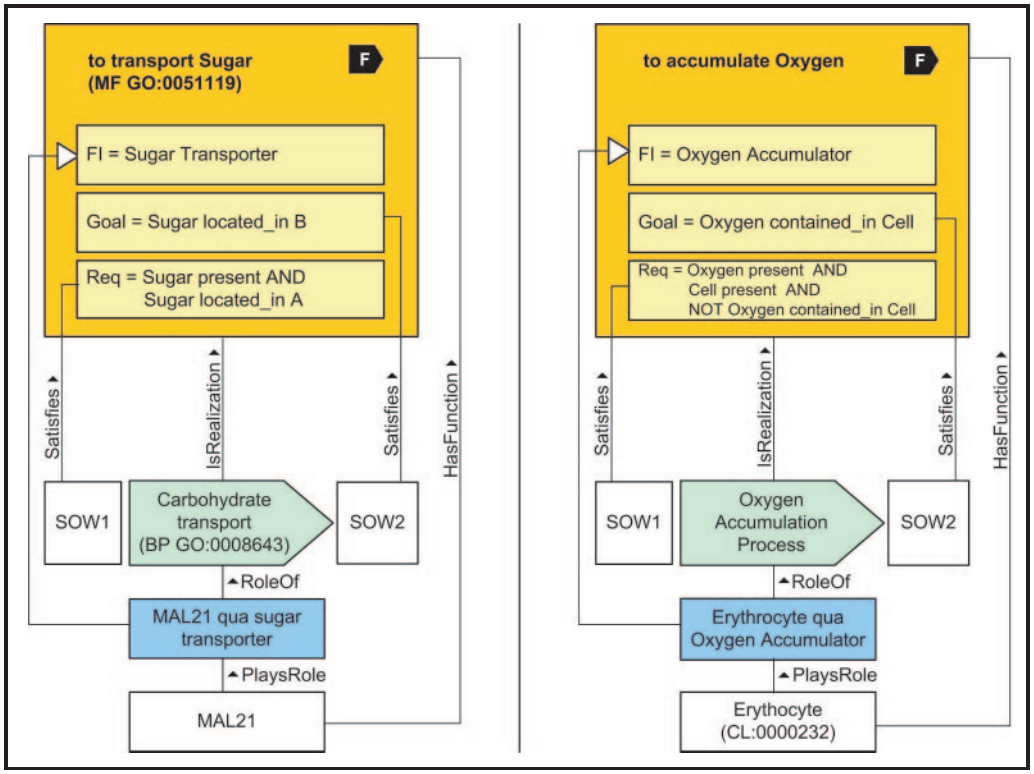


Fig. 2. Two exemplary models employing OF, instantiating the general model in Figure 1 (correspondences indicated by the coloring). On the left-hand side, a schematic version of the function ''to transport sugar'' together with its realization is shown. Processes of type ''carbohydrate transport'' realize this function, and an entity, in this case MAL21, has the function ''to transport sugar''. Whenever applicable, the identifiers from the GO are used (for the function and process). MAL21 is currently annotated to the function and the process in the GO. In this model, the annotation relation is replaced by the *HasFunction* relation. On the right-hand side, the function "to accumulate oxygen" is modelled. This is a function taken from the Celltype Ontology. Except for erythrocyte, the entities involved in this model are not present in any of the OBO ontologies.

So let us investigate how the function ''to transport sugar'' can be modelled in the framework of the OF.

- As requirements, we assume that a sugar-molecule (CHEBI:25407 or CHEBI:25679) is located at some location.
- The goal is the location of the sugar molecule at a different location.
- The functional item is a role which we call ''sugar transporter''.

We find that many of the gene products annotated with the ''sugar transporter activity'' in GO's Molecular Function taxonomy are also annotated with some sub-category of the ''transport'' (GO:0006810) or ''carbohydrate transport'' (GO:0008643) categories in GO's Biological Process taxonomy.

Also the names of the categories indicate a link, and of course there is an obvious one: gene products which have the function ''to transport'' may participate in a ''transport'' process. With the help of OF, we can make explicit some links between categories in GO's Molecular Function and Biological Process taxonomies: Processes of type ''carbohydrate transport'' (GO:0008643) are realizations of the function ''to transport sugar''; many of the gene products annotated with either carbohydrate transport or sugar transporter activity, such as MAL21 (maltose permease), can stand in the *HasFunction* relation to ''to transport sugar''; new categories appear, namely gene products acting as (or ''qua") transporter, e.g. MAL21 *qua* transporter.

The left-hand side of Figure 2 demonstrates the full interconnections of this example by means of OF. In terms of the relations we introduced this is captured by *Realizes*(MAL21, GO:0051119, GO:0008643). What could be directly added to the GO are links of *IsRealization* and *HasFunction*: *IsRealization*(GO:0008643, GO:0051119) and *HasFunction*(MAL21, GO:0051119).

However, considering the GO's definition of ''sugar transporter activity'' Enables the directed movement of a sugar into, out of, within or between cells. A sugar is any member of a class of sweet, water-soluble, crystallizable carbohydrates, which are the monosaccharides and smaller oligosaccharides.

It is possible to interpret this function differently: as the function ''to enable F'' or ''to support F'', where F is the function ''to transport sugar''.

Now the function ''to support F'' with F being ''to transport sugar'' would simply be a function where the goal of ''to support F'' would be part of the requirements of ''to transport sugar". So every realization of ''to support F'' would be a transition from a state of the world where some of the requirements for ''to transport sugar'' (the presence of a sugar molecule or its location) are not satisfied to a state where they are satisfied.

Many more relations between functions can be modelled and may be relevant in GO, such as ''to trigger'' or ''to prevent". Separating

these functions, which is made possible using OF, could lead to more accurate and comprehensive definitions.

2.6.2 Identifying implicit functions and processes The Ontology of Functions can be applied to existing taxonomies in order to make explicit functions and processes which are currently implied but not separately defined.

This kind of use of the concept of function occurs in the Celltype Ontology (Bard *et al*., 2005) (CL) and the Ontology of Chemical Entities of Biological Interest (Brooksbank *et al*., 2005) (ChEBI). We will only explore the Celltype Ontology, but the same argument can be applied to ChEBI.

CL uses the term function in the subtree ''cell by function'' which classifies cell types by the functions which they perform. A general example is ''stuff accumulating cell'' (CL:0000325), and more specifically ''oxygen accumulating cell'' (CL:0000329), of which a red blood cell or erythrocyte (CL:0000232) is a subcategory. The function ''to accumulate oxygen (by a cell)'' would be modelled as shown in the right-hand side of Figure 2:

- The presence of oxygen (ChEBI:25805) outside of a cell (CL:0000000) is the requirement of the function.
- The goal of the function is the cell's accumulation of oxygen: The oxygen is contained in the cell.
- The functional item is called ''oxygen accumulator''.

The subsumption of erythrocyte under oxygen accumulating cell in CL reflects the fact that erythrocytes have the function ''to accumulate oxygen'', *HasFunction*(CL:0000232, ''to accumulate oxygen''). Further, they may *act as* oxygen accumulators, a new category for CL, in the process of ''oxygen accumulation'', *IsRealization* (''oxygen accumulation'', ''to accumulate oxygen''). Again, the *Realizes* relation captures all these new relations appropriately: *Realizes*(CL:0000232, ''to accumulate oxygen'', ''oxygen accumulation'').

The analysis of erythrocyte in CL has led to the discovery of entities which are not yet part of CL or any other OBO ontology, but which contribute to the understanding of interactions among ontologies in cellular biology. Additionally, we can now define ''oxygen accumulating cell'' as a cell which has the function ''to accumulate oxygen''.

3 DISCUSSION

3.1 Adding information systematically

The framework developed here and fully described in (Burek, 2006) can be used to provide additional information for existing biomedical ontologies such as the Gene Ontology (GO), without the need for modification of the existing structure of these ontologies. In general, we provide a *methodology* for defining functions and relating them to various other entities, such as processes, roles and even genes and gene products. This methodology may benefit the annotation and curation process and lead to improved definitions and completeness.

The advantage of the Ontology of Functions (OF) is *enhanced expressivity*. For example, the curators of the GO when annotating a gene product with the appropriate terms from the GO will have the information available that a certain protein is involved in some process and how it is related to a certain molecular function. They may also have more information about the protein, for example the conditions under which it operates and other requirements which need to be satisfied for the protein to be active. By means of the OF, this information can be made explicit, and will not be lost as is currently the case.

The OF further allows for a refinement or *replacement of the annotation relation* in a number of cases by means of the *Has Function* relation. Note that the latter is an ontological relation, in contrast to the annotation relation, which is currently an arbitrary association relation introduced to link genes and gene products to the concepts of an ontology. Refined annotations do not only provide more information within ontologies themselves, but also with respect to the relation between categories of biomedical ontologies and genomic knowledge about biological reality.

Both, additional information due to enhanced expressivity as well as refined annotations may prove useful for the various statistical methods which have been applied to biomedical ontologies in order to detect biological correlations, such as (Subramanian *et al*., 2005; Beissbarth and Speed, 2004; Berriz *et al*., 2003).

It is interesting to consider to what extent and how the addition of information to existing biomedical ontologies can be automated. At present, we do not have an implemented solution for this issue. However, we expect that approaches to finding associations between categories using lexical and statistical analysis like (Bodenreider *et al*., 2005; Burgun *et al*., 2004) can be exploited and combined with the OF, in order to add categories and relations between them automatically. These could further be verified by existing natural language processing techniques (Mungall, 2004).

However, the rich formalism of the OF introduces another kind of new information which is less likely to be added automatically: *roles* and *qua-individuals*, the instances of roles. These concepts have mostly been neglected in the bio-ontology community, but ontological research has dealt with roles for a long time and rich theories of roles exist (Guarino andWelty, 2000, 2004; Masolo *et al*., 2004, 2005; Poli, 1998; Loebe, 2003, 2005). We believe that they can prove useful in the explanation of biological phenomena. Making them explicit in biomedical ontologies can therefore serve to complete the coverage of these ontologies and enhance their conceptual modelling capabilities.

However, ontological theories must be applied cautiously. For instance, the theory of roles as proposed in (Guarino and Welty, 2004) defines constraints on the subsumption relation. Applied to an example from the Celltype Ontology, the subsumption link between ''red blood cell'' and ''circulating cell'' violates that constraint, if ''circulating cell'' is understood as a role. In this case ''circulating cell'' would refer to the role played by a red blood cell in the actual process of circulation. We, on the other hand, analyze ''circulating cell'' as a cell which has the actual or dispositional function ''to circulate'', which would not violate a subsumption constraint in (Guarino and Welty, 2004).. This said, we want to emphasize that the application of formal ontological theories to domain ontologies must be done cautiously, and preferably in collaboration with domain experts.

3.2 Automated reasoning

The relation of our proposal to *automated reasoning* is highly relevant in the context of biomedical ontologies. Automated reasoning on biological data has been a goal of the bioinformatics and the bio-ontology community for some time (Wroe *et al*., 2003). We believe

that much benefit can be gained from automated reasoning if a rich set of axioms is provided. The Ontology of Functions is equipped with a rich axiomatization (see (Burek, 2006)), which can be—for reasons of efficiency—adapted to description logic and used in conjunction with an automated reasoner such as FaCT (Horrocks *et al*., 1999).

Therefore, the OF can be seen as a formal and unambiguous specification framework for biological functions whose consistency can be verified, and in which implicit knowledge can be deduced.

3.3 Related work

To our knowledge, the only approach which in its aim is strictly similar to our proposal is that of Karp (2000). This proposal, however, is limited to a molecular granularity. Biological functions on the cellular, organismal or population level of granularity are not included. Moreover, functions are explicitly not context-dependent, while in the OF the *has-function* relation can, in principle, be dependent on a context. Furthermore, (Karp, 2000) attempted to create an ontology of functions as a module for EcoCyc[4]. The OF, on the other hand, is a top-level ontology of functions, and is therefore domain-independent and general. However, the view which (Karp, 2000) takes on functions is compatible with the OF.

The Gene Ontology (Ashburner *et al*., 2000) also provides a definition for a molecular function:

> Molecular function describes activities, such as catalytic or binding activities, at the molecular level. GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the actions, and do not specify where or when, or in what context, the action takes place.

However, this definition does not separate activities and functions, as is the case in the OF which distinguishes functions and their realizations. Adding this distinction allows the capture of more information in the GO, while retaining GO's current structure.

In philosophy and ontology, many theories about biological functions have been developed (Searle, 1995; Johansson, 2004; Johansson *et al*., 2005; Kumar and Smith; Millikan, 1987; Melander, 1997). However, while these discussions provide valuable theoretical insight, they do not provide an immediate practical solution to the problem of conceptual modelling of functions in biology. We tried to learn from these discussions and develop the means for modelling function.

Many attempts to integrate the taxonomies of the GO have been made (Hill *et al*., 2002; Kumar *et al*., 2004; Aranguren, 2004; Wroe *et al*., 2003; Aranguren, 2005). However, none of these are based on a thorough ontological analysis of functions and their relation to other relevant biological entities such as processes.

4 CONCLUSION

The Ontology of Functions provides a framework for representing arbitrary functional knowledge in every domain of biology. This framework is used to define and specify functions, and relate them to other entities in biology. This helps to prevent errors, to clarify definitions and to support the integration of biological data and knowledge. We have shown how to use the OF to represent the relation between biological processes and functions in the Gene Ontology, for which no ontologically founded representation formalism is currently available.

The introduced formalism requires no changes to the existing structure of the Gene Ontology, and could therefore be adopted gradually. Moreover, we have demonstrated how to analyze the annotation relation in the OF. Based on such analyses, the relation between genes or gene products and categories to which they are annotated can be made more precise. We have further shown how the OF framework can be used to identify and define functions of cells or chemicals.

The OF is a top-level ontology of functions which will be extended by including biological domain concepts. Statistical methods or text mining methods such as (Bodenreider *et al*., 2005; Burgun *et al*., 2004) could be used to extract the skeleton of a functional domain ontology from the existing ontologies. The OF can also be used to support the construction of a biological core ontology, which is a top-level ontology for the domain of biology (cf. also (Rector *et al*., 2006)) for an initial proposal of such an ontology).

Moreover, we are working on an implementation of this framework in the form of an annotation and curation tool, which will effectively guide the annotation and curation process by implementing the methodology defined by the OF to represent functional knowledge. We plan to integrate an automated reasoner with this tool in order to assist in maintaining consistency and to enable automated deduction.

[4]http://ecocyc.org/

ACKNOWLEDGEMENTS

We thank Michael Lachmann, Katrin Loebe, and Kay Prüfer for helpful discussions and critical reading of the manuscript. We thank the Max Planck Society, the German Federal Ministry of Education and Research, the Institute of Medical Informatics, Statistics and Epidemiology, and the Graduiertenkolleg Knowledge Representation of the German Research Foundation for financial support.

REFERENCES

V. Akman and M. Surav. Steps toward formalizing context. *AI Magazine*, 17(3):55–72, 1996.

M. E. Aranguren. Improving the structure of the gene ontology. Master's thesis, University of Manchester, 2004.

M. E. Aranguren. Ontology design patterns for the formalisation of biological ontologies. Master's thesis, University of Manchester, 2005.

M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, May 2000.

J. Bard, S. Y. Rhee, and M. Ashburner. An ontology for cell types. *Genome Biol*, 6(2):R21, 2005.

T. Beissbarth and T. P. Speed. GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics*, 20(9):1464–1465, Jun 2004.

G. F. Berriz, O. D. King, B. Bryant, C. Sander, and F. P. Roth. Characterizing gene sets with FuncAssociate. *Bioinformatics*, 19(18):2502–2504, Dec 2003.

O. Bodenreider, M. Aubry, and A. Burgun. Non-lexical approaches to identifying associative relations in the gene ontology. *Pac Symp Biocomput*, pages 91–102, 2005.

C. Brooksbank, G. Cameron, and J. Thornton. The European Bioinformatics Institute's data resources: towards systems biology. *Nucleic Acids Res*, 33(Database issue): D46–D53, Jan 2005.

P. Burek. *Ontology of Functions*. PhD thesis, University of Leipzig, Institute of Informatics (IfI), 2006. forthcoming.

A. Burgun, O. Bodenreider, M. Aubry, and J. Mosser. Dependence relations in gene ontology: A preliminary study. In *Proceedings of the Workshop on The Formal Architecture of the Gene Ontology*, 2004.

N. Guarino and C. A. Welty. A formal ontology of properties. In *Lecture Notes in Computer Science*, volume 1937, pages 97–112, 2000.

N. Guarino and C. A. Welty. An overview of OntoClean. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, chapter 8, pages 151–172. Springer, 2004.

B. Heller and H. Herre. Ontological categories in GOL. *Axiomathes*, 14(1):57–76, 2004a.

B. Heller and H. Herre. Ontological foundations of medical information systems. In H. Fujita and V. Gruhn, editors, *New Trends in Software Methodologies, Tools and Techniques: Proceedings of the Third SoMeT_W04*, volume 111 of *Frontiers in Artificial Intelligence and Applications*, pages 3–17, Amsterdam, 2004b. IOS Press.

B. Heller, H. Herre, K. Lippoldt, and M. Loeffler. Standardized terminology for clinical trial protocols based on top-level ontological categories. In K. Kaiser, S. Miksch, and S. Tu, editors, *Computer-based Support for Clinical Guidelines and Protocols: Proceedings of the Symposium on Computerized Guidelines and Protocols,* volume 101 of *Studies in Health Technology and Informatics*, pages 46–60. IOS Press, 2004.

H. Herre and B. Heller. Ontology of time and situoids in medical conceptual modeling. In S. Miksch, J. Hunter, and E. T. Keravnou, editors, *Proceedings of the 10th Conference on Artificial Intelligence in Medicine (AIME 05), Aberdeen, Scotland, Jul 23–27*, volume 3581 of *Lecture Notes in Computer Science*, pages 266–275, Berlin, 2005. Springer.

H. Herre, B. Heller, P. Burek, R. Hoehndorf, F. Loebe, and H. Michalek. General Formal Ontology (GFO) – a foundational ontology integrating objects and processes. Technical report, University of Leipzig, 2006.

D. P. Hill, J. A. Blake, J. E. Richardson, and M. Ringwald. Extension and integration of the gene ontology (GO): combining GO vocabularies with external vocabularies. *Genome Res*, 12(12):1982–91, 2002.

I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proceedings of LPAR'99*, LNCS, 1999.

Y. Iwasaki and B. Chandrasekaran. Design verification through function- and behavior-oriented representations. In J. S. Gero and F. Sudweeks, editors, *Proceedings Artificial Intelligence in Design Conference*, pages 597–616, 1992.

I. Johansson. Functions, function concepts, and scales. *The Monist*, 87(1):96–114, 2004.

I. Johansson, B. Smith, K. Munn, N. Tsikolia, K. Elsner, D. Ernst, and D. Siebert. Functional anatomy: A taxonomic proposal. *Acta Biotheoretica*, 53:153–166, 2005.

P. D. Karp. An ontology for biological function based on molecular interactions. *Bioinformatics*, 16(3):269–285, Mar 2000.

A. Kumar and B. Smith. The ontology of processes and functions: A study of the international classification of functioning, disability and health. http://ontology.buffalo.edu/medo/ICF.pdf. Draft.

A. Kumar, B. Smith, and C. Borgelt. Dependence relationships between gene ontology terms based on TIGR gene product annotations. In *Proc CompuTerm*, pages 31–38, 2004.

F. Loebe. An analysis of roles: Towards ontology-based modelling. Master's thesis, University of Leipzig, Institute of Informatics (IfI), 2003.

F. Loebe. Abstract vs. social roles: A refined top-level ontological analysis. In G. Boella, J. Odell, L. van der Torre, and H. Verhagen, editors, *Proceedings of the 2005 AAAI Fall Symposium Roles, an Interdisciplinary Perspective: Ontologies, Languages, and Multiagent Systems', Nov* 3–6, *Arlington, Virginia*, number FS-05–08 in Fall Symposium Series Technical Reports, pages 93–100, Menlo Park (California), 2005. AAAI Press.

C. Masolo, G. Guizzardi, L. Vieu, E. Bottazzi, and R. Ferrario. Relational roles and qua-individuals. In G. Boella, J. Odell, L. van der Torre, and H. Verhagen, editors, *Proceedings of the 2005 AAAI Fall Symposium 'Roles, an Interdisciplinary Perspective: Ontologies, Languages, and Multiagent Systems', Nov 3–6, Arlington, Virginia*, number FS-05–08 in Fall Symposium Series Technical Reports, pages 103–112, Menlo Park (California), 2005. AAAI Press.

C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino. Social roles and their descriptions. In D. Dubois, C. Welty, and M.-A. Williams, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004), Whistler, Canada, June 2–5*, pages 267–277, Menlo Park, 2004. AAAI Press.

J. McCarthy and S. Buvač. Formalizing context (expanded notes). In A. Aliseda, R. J. van Glabbeek, and D. Westerståhl, editors, *Computing Natural Language*, volume 81 of *CSLI Lecture Notes*, pages 13–50. Center for the Study of Language and Information (CSLI), Stanford University, Stanford, 1998.

P. Melander. *Analyzing Functions. An Essay on a Fundamental Notion in Biology*. Almqvist and Wiksell, 1997.

R. G. Millikan. *Language, Thought, and Other Biological Categories: New Foundations for Realism*. The MIT Press, 1987.

C. J. Mungall. Obol: integrating language and meaning in bio-ontologies. *Comp Funct Genomics*, 5:509–520, 2004.

OMG. Unified Modeling Language: Infrastructure. Specification v2.0, Object Management Group (OMG), Needham (Massachusetts), Mar 2006. http://www.omg.org/docs/formal/05-07-05.pdf.

R. Poli. Qua-theories. In L. Albertazzi, editor, *Shapes of Forms*, pages 245–256. Kluwer, 1998.

A. Rector, R. Stevens, and J. Rogers. Simple bio upper ontology. http://www.cs.man.ac.uk/∼rector/ontologies/simple-top-bio/, 2006.

M. A. Rosenman and J. S. Gero. Purpose and function in design. *Design Studies*, 2:161–186, 1998.

M. Sasajima, Y. Kitamura, M. Ikeda, and R. Mizoguchi. FBRL: A function and behavior representation language. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI'95), Montréal, Québec, Canada, Aug 20–25*, volume 2, pages 1830–1836. Morgan Kaufmann, 1995.

J. R. Searle. *The Construction of Social Reality*. Penguin Group, 1995.

B. Smith, W. Ceusters, B. Klagges, J. Köhler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. L. Rector, and C. Rosse. Relations in biomedical ontologies. *Genome Biol*, 6(5):R46, 2005.

B. Smith, J. Williams, and S. Schulze-Kremer. The ontology of the gene ontology. *AMIA Annu Symp Proc*, pages 609–613, 2003.

A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci USA*, 102(43): 15545–15550, Oct 2005.

C. J. Wroe, R. Stevens, C. A. Goble, and M. Ashburner. A methodology to migrate the gene ontology to a description logic environment using DAML+OIL. *Pac Symp Biocomput*, pages 624–635, 2003.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e74–e80
doi:10.1093/bioinformatics/btl215

Comparative footprinting of DNA-binding proteins

Bruno Contreras-Moreira[1,*] and Julio Collado-Vides[1]

[1]Programa de Genómica Computacional, Centro de Ciencias Genómicas, Universidad Nacional Autónoma de México, Av.Universidad, s/n, 62210 Cuernavaca, Morelos, México

ABSTRACT

Motivation: Comparative modelling is a computational method used to tackle a variety of problems in molecular biology and biotechnology. Traditionally it has been applied to model the structure of proteins on their own or bound to small ligands, although more recently it has also been used to model protein-protein interfaces. This work is the first to systematically analyze whether comparative models of protein-DNA complexes could be built and be useful for predicting DNA binding sites.

Results: First, we describe the structural and evolutionary conservation of protein-DNA interfaces, and the limits they impose on modelling accuracy. Second, we find that side-chains from contacting residues can be reasonably modeled and therefore used to identify contacting nucleotides. Third, the DNASITE protocol is implemented and different parameters are benchmarked on a set of 85 regulators from *Escherichia coli*. Results show that comparative footprinting can make useful predictions based solely on structural data, depending primarily on the interface identity with respect to the template used.

Availability: DNASITE code available on request from the authors

Contact: contrera@ccg.unam.mx

Supplementary information: http://www.ccg.unam.mx/Computational_Genomics/supplementary/ismb2006

1 INTRODUCTION

Comparative modelling is now a mature technology that predicts the three-dimensional arrangement of a protein sequence given an alignment to one or more template proteins of known structure. The use of protein models may range from site-directed mutagenesis and molecular replacement to molecular docking and protein design and engineering (Baker and Sali, 2001; Contreras-Moreira *et al*., 2002). The actual use of a protein model will depend on its expected accuracy, dictated primarily by the sequence similarity to the templates used (Contreras-Moreira *et al*., 2005; Chothia and Lesk, 1986). Together with sequence alignment errors, this is a main factor affecting model quality (Tramontano *et al*., 2001). This factor has also been found to be critical when reconstructing protein-protein interfaces (Aloy *et al*., 2003); the more similar the sequences, the more predictable the details of the interface.

*To whom correspondence should be addressed.

In this paper we ask these questions to a different system, the interface between proteins and nucleic acids. There has been great interest in understanding these interactions, given the biological relevance of genetic regulation (Sarai and Kono, 2005). For this reason a good amount of experimental work has been dedicated to this problem, most of it now part of the Protein Data Bank (PDB) (Berman *et al*., 2000). This work takes all this experimental data, i.e. crystallographic and NMR structures, in order to:

(1) determine if there are any evolutionary trends which might explain the divergence of protein-nucleic acid interfaces and therefore support comparative modelling of these complexes

(2) assess if footprinting predictions can be made by comparative modelling of protein-DNA complexes

The motivation for this analysis stems from a variety of approaches recently tested on experimentally determined complexes, that isolate and characterize the preferred recognised sequences of transcription factors by using physical (Aloy *et al*., 1998; Gromiha *et al*., 2005; Kono and Sarai 1999; Luscombe *et al*., 2001; Morozov *et al*., 2005; Nadassy *et al*., 1999; Pabo and Nekludova 2000; Paillard and Lavery 2004; Selvaraj *et al*., 2002; Siggers *et al*., 2005; Steffen *et al*., 2002) and evolutionary metrics (Kaplan *et al*., 2005; Raviscioni *et al*., 2005). Here we demonstrate that comparative modelling can help explain or predict the repertoire of known binding sites of a given regulator, annotated in resources such as RegulonDB (Salgado *et al*., 2006), for proteins for which no structural description is available, provided that we know the structure of homologous proteins.

This work presents the first systematic benchmark of comparative modelling protein-DNA complexes with the aim of predicting DNA operator sites. First we compile a non-redundant set of protein-DNA complexes to assess the conservation of their interfaces. The results show that comparative modelling of these complexes is possible with one restriction: as sequence similarity diminishes protein-DNA interfaces diverge exponentially. Second we implement a protocol that we call DNASITE that builds comparative models of protein-DNA interfaces using tools and datasets widely used by the structural bioinformatics community. Finally we choose the appropriate parameters and test the performance of DNASITE on a set of 85 *Escherichia coli* regulator proteins for which RegulonDB contains known binding-sites with experimental evidence.

2 METHODS

Collecting protein-DNA complexes

We retrieved all PDB entries (as of August 9, 2005) containing both protein and DNA coordinates, and selected all protein chains less than 12Å away from any DNA segment. This list of chains was pruned using a 95% sequence identity cut-off to get a non-redundant set, using the web server PISCES (Wang and Dunbrack, 2003). We then put every selected chain together with the contacting nucleic acid molecules and called that a PN complex, where P stands for protein and N for nucleic acid. The resulting library contained 273 crystallographic and NMR structures and is available as supplementary material.

Comparing complexes by means of protein structural alignments

The next step of our procedure was to compare the protein chains of all complexes using structural alignments, as a way of minimizing possible alignment errors. For this we used the program MAMMOTH (Ortiz *et al*., 2002) and considered only pairs of complexes that yielded –ln(E) values over 4.5 and had at least 10% of sequence identity, to eliminate non statistically significant matches. From more than 37000 comparisons, 442 passed this filter and were used to plot the conservation of protein-nucleic acid interfaces as sequence similarity changed. Each of these pairs resulted in a structural superposition with an associated sequence alignment. Eight folds from the Structural Classification of Proteins (SCOP) (Murzin *et al*., 1995) dominate this dataset, as shown in Results.

Calculating interface agreement between superposed complexes

For each complex pair (A,B) we calculated three numbers: the sequence identity (ID_{ab}) between protein chains P_a and P_b; the structural agreement of the amino acid residues participating in the interface (P-$RMSD_{ab}$); and the structural agreement of the interface nucleotides (N-$RMSD_{ab}$). Calculating ID_{ab} is simple, matches in the sequence alignment divided by the total number of aligned residues. The other two numbers are calculated from the structural superposition of PN_a over PN_b in six steps:

(1) P_a residues contacting N_a nucleotides are put in set P_{ac}.

(2) P_b residues aligned to those in P_{ac} are put in P_{bc}.

(3) Residues in P_{ac} and P_{bc} are taken in pairs to calculate their root-median-square deviation. We call this number P-$RMSD_{ab}$.

(4) For each residue in P_{ac}: closest nucleotide in N_a is put in set N_{ac}.

(5) For each residue in P_{bc}: closest nucleotide in N_b is put in set N_{bc}.

(6) Nucleotides in N_{ac} and N_{bc} are taken in pairs to calculate their root-median-square deviation. We call this number N-$RMSD_{ab}$.

Protein residues were represented by their C_α atoms, while for nucleotide bases we took N9 (purines) and N1 (pyrimidines) atoms. For step 1, a protein-nucleic acid contact is defined as a pair of atoms placed less than 12Å away from each other, following the work of Aloy *et al.* (Aloy *et al*., 1998). For step 2 we require aligned protein residues to be within 4Å from each other after superposition.

Calculating side-chain modelling accuracy

1477 H-bonding residues from our library of superposed complexes were modelled with the program SCWRL2.7 (Dunbrack and Karplus, 1993) and RMSD values were calculated for each model-experimental pair of side-chains. For each pair(A,B), first A was used as template to predict B side-chains and then B was chosen as template.

Table 1. Protein-DNA recognition matrix compiled by the authors (CM parameter set) from a set of 273 95% non-redundant complexes. Contacts were identified using a distance threshold of 4Å (from any side-chain atom to any atom in the purine/pyrimidine ring). Each value is a log-odd calculated as in (Mandel-Gutfreund, *et al*., 2001)

	C	G	A	T
D	+0.26	−0.49	−1.79	−1.11
P	−1.31	−1.81	−0.73	−0.15
I	−1.06	−1.64	−0.53	−0.99
K	−0.54	+1.05	−0.75	+0.35
W	+0.44	+0.34	−0.47	+0.07
C	−0.74	−1.83	−0.85	−0.36
G	−2.57	−2.57	−2.57	−2.57
F	−0.76	+0.01	+0.06	+0.30
Q	+0.21	+0.49	+0.63	+0.25
S	−0.40	+0.42	−0.50	+0.62
N	+0.41	+0.46	+0.98	+0.65
L	−1.76	−1.29	−1.03	−0.65
V	−0.97	−2.57	−0.43	−0.06
E	+0.53	−1.65	−1.62	−1.09
Y	+0.55	+0.60	+0.36	+0.88
R	+0.76	+1.96	+0.56	+1.09
T	+0.26	−0.35	−0.41	+0.44
M	−0.40	+0.31	+0.10	+0.39
A	−1.10	−1.31	−1.21	−0.27
H	−0.39	+1.01	−0.49	+0.54

Implementation of DNASITE

The DNASITE protocol was programmed in Perl and C and is conceptually very simple. The input is a protein sequence and these are the steps that follow:

(1) Search for homologous protein-DNA complexes with three iterations of PSI-BLAST (Altschul *et al*., 1997), using a sequence library made of the proteins in our non-redundant set of complexes plus the sequences in SWISSPROT (Sep, 2005) (Bairoch and Apweiler, 2000).

(2) Use local PSI-BLAST alignments to build the protein backbone of the modelled complex, using the template's coordinates. Accept only models that align residues known to be contacting nucleotides in the template.

(3) Add SCWRL side-chains keeping the template DNA in frame. We can choose to model only mutated side-chains.

(4) Identify binding residues as those less than 4.5Å away from any atom in the purine/pyrimidine ring, a similar distance to that used previously by Mandel-Gutfreund (Mandel-Gutfreund and Margalit, 1998). These residues are used to calculate the % interface identity (IID).

(5) Thread DNA sequences into the modelled complex and evaluate the matching using logarithmical protein-DNA 20x4 recognition matrices, such as those derived by Mandel-Gutfreund (Mandel-Gutfreund *et al*., 2001). The scoring function (Equation 1) is additive, assuming that each residue in the interface contributes equally to the matching score. A family-specific correction might be applied, calculating a correction term derived from the background substitution frequencies contained in the PSI-BLAST position-specific scoring matrices (PSSM) and the protein-DNA matrix used, as described in Equation 2. The idea is that amino acid substitutions might be indicating which nucleotide bases are preferred at each position, somehow capturing context-dependent preferences. DNA deformation for each

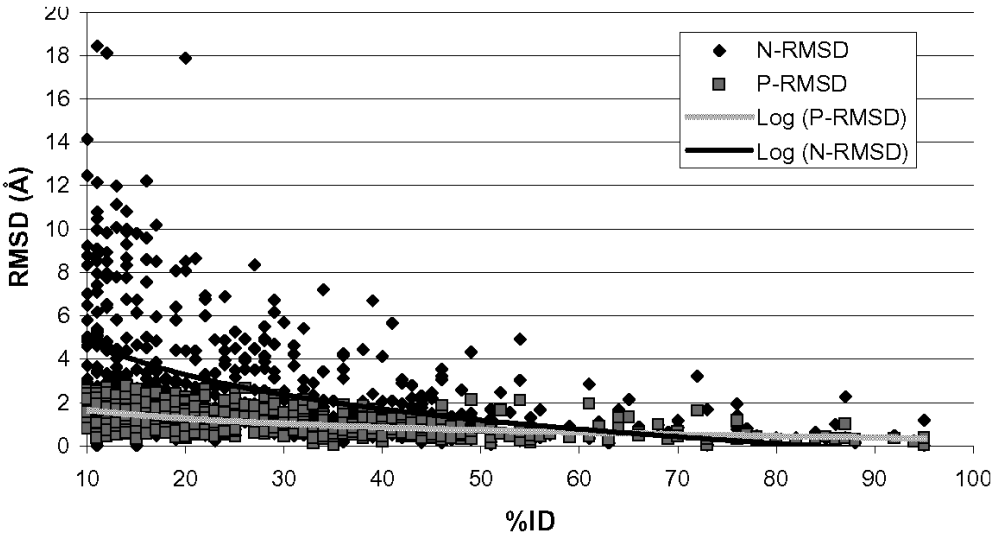


Fig. 1. Interface conservation in terms of P-RMSD and N-RMSD. 442 pairs of protein-nucleic acid complexes were superposed and the conservation of their interfaces plotted against their protein sequence identity. Two measures are reported: P-RMSD, the median deviation of the protein residues taking part in the interface; N-RMSD, the median deviation of the nucleotides of the interface. Logarithmical regression lines are added to assist in the interpretation.

threaded sequence is approximately estimated using the X3DNA package (Lu and Olson, 2003), in order to consider also indirect readout mechanisms (Gromiha *et al*., 2005). Briefly, DNA parameters (step, shift, slide, rise, tilt, roll, twist) are calculated from the template DNA molecule and then used to approximate deformation energies based on sequence-dependent parameters (Olson *et al*., 1998) (Marc Parisien, personal communication). The native DNA molecule is used as a reference and an arbitrary cut-off is set to skip sequences with large deformation energies. To ensure fast computation times, shortcuts are applied when the number of possible DNA sequences is greater than 4^9. Only the top fraction of sequences is selected to build a footprinting matrix. If the number of selected sequences is less that 50 the DNA sequence of the template complex is added.

Given a PN complex, with L interface nucleotides contacting C protein residues and a scoring matrix, the scoring function is calculated as follows:

$$Score(PN) = \sum_{i=1}^{L}\sum_{j=1}^{C} match\ (P_i, N_j, matrix) \quad (1)$$

To calculate the family correction for a given residue P_j in contact with nucleotide base N_i, each of the 20 possible aminoacid (aa) substitution frequencies in a PSSM are considered:

$$Corr(P_j, N_i) = \sum_{x=1}^{20} freq(aa(x))match(aa(x), N_i, matrix) \quad (2)$$

DNASITE benchmark

The set of known and putative regulator proteins in *E.coli* was taken as a test set, including 3 SCOP folds. Each of those sequences was used as input for DNASITE and 85 comparative models were obtained (IHF was excluded from this test as it was considered to be non-sequence specific). Each of these 85 models was built using different parameters that will be referred to using these codes:

- Def: default parameters, using a 2001 Mandel-Gutfreund matrix, up to three contacts per residue and a DNA deformation cut-off of 1.6 kcal/mol.
- CM: uses a matrix built by the authors from the non-redundant set of complexes, based only on distance cut-offs (see Table 1).
- Sc3: uses SCWRL3.0 (Canutescu *et al*., 2003), instead of version 2.7, to compare the performance.
- Df1: uses a DNA deformation energy cut-off of 1 kcal/mol.
- Df2: uses a DNA deformation energy cut-off of 2 kcal/mol.
- Df3: uses a DNA deformation energy cut-off of 3 kcal/mol.
- C1: only one contact per residue is considered, the closest one.
- M: conservative, models only mutated side-chains, the rest are taken as in the template complex.
- F: uses family-specific correction.
- P: P-value cut-off for selecting threaded sequences.

The footprint matrices generated by DNASITE were aligned against the corresponding set of known binding sites extracted from RegulonDB (Jan, 2006) using the program PATSER (Hertz and Stormo, 1999). Each site is flanked by segments of 10 nucleotides. Alignments yielding significant scores, over the cut-off estimated by PATSER for each matrix, were considered as recovered sites and for those the average ln(P-value) was calculated. Finally, the aligned sites were used to build a sequence logo with WebLogo (Crooks *et al*., 2004).

3 RESULTS

3.1 Protein-DNA interface conservation

Figure 1 shows N-RMSD and P-RMSD values obtained from a total of 442 non-redundant complex superpositions plotted against %ID. Individual N-RMSD and P-RMSD data points are depicted and logarithmic regression lines are added to help interpretation. Note that interface nucleotides accumulate larger deviations when superposed than their contacting residues. Furthermore, both N-RMSD and P-RMSD are significantly correlated to %ID, with correlation coefficients of −0.43 and −0.52 respectively. Nucleotide median deviations for complexes with at least 30% of sequence identity tend to be close to 2Å, more precisely within the 1.4 ± 1.2Å interval.

As mentioned earlier, 8 SCOP folds are over-represented in our dataset, the most common being the DNA/RNA binding 3-helical

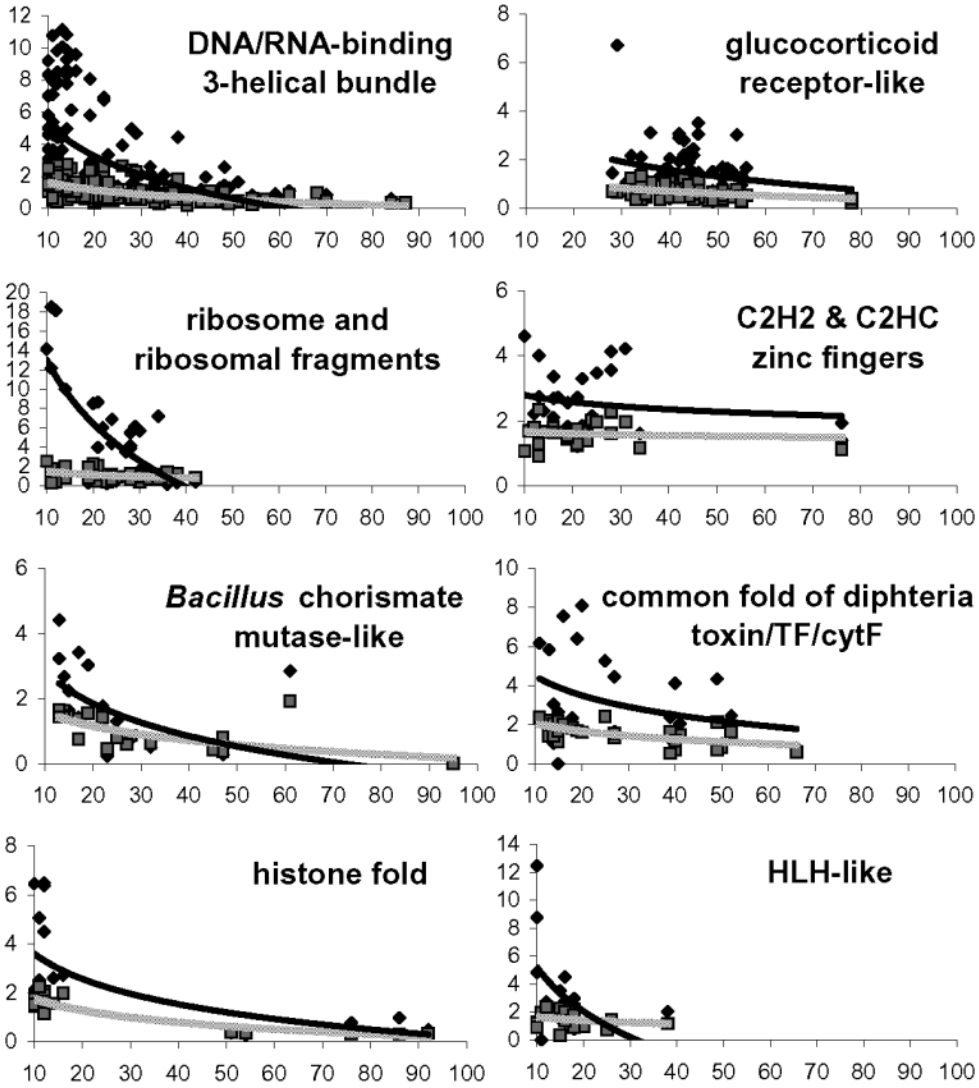


Fig. 2. Interface conservation for 8 representative SCOP folds. Same analysis as in Figure 1, splitting the data corresponding to the most abundant SCOP folds in our dataset. For all panels X-axis is %ID and Y-axis is RMSD measured in Å, with N-RMSD plotted in black and P-RMSD in grey. A majority of *E.coli* transcription factors contain helix-turn-helix motifs and can be classified as DNA/RNA-binding 3-helical bundle folds.

bundle. Figure 2 shows the same analysis performed on these most abundant SCOP folds, showing more specific trends, as also noticed by Siggers (Siggers *et al*., 2005).

These results are encouraging as they indicate that interfaces are structurally and evolutionary related and their sequence similarity is a reasonable estimator of the degree of conservation. However, before we can build comparative models of these complexes we need to previously identify which modelled amino acid residues are contacting DNA bases.

3.2 Side-chain modelling accuracy

In order to identify which residues are contacting nucleotides in a complex we first need to model the residue side-chains. As explained in Materials and Methods, we used the program SCWRL2.7 for this task and found that 77% of H-bonding modelled side chains deviate less than 2.0Å in average with respect to the experimental coordinates, excluding pairs of complexes with less than 30% sequence identity. We concluded that we can reasonably predict side-chain rotamers and therefore which residues are likely contacting nucleotides.

3.3 Footprinting of comparative protein-DNA complexes

Table 2 shows the performance of the DNASITE protocol using our test set of 85 *E.coli* regulators, comprising three folds: DNA/RNA-binding 3-helical bundles, lambda repressors and Met repressors. Three measurements are taken for each run: the percentage of recovered sites, the mean alignment score and the mean significance of alignment scores. This benchmark highlights some parameters settings, those that perform well in recovering RegulonDB sites with significant scores. Three of them were selected, P0.0001, MF and FP0.0001, and a few representative examples of footprinting predictions are shown in Figure 3. What do these parameters

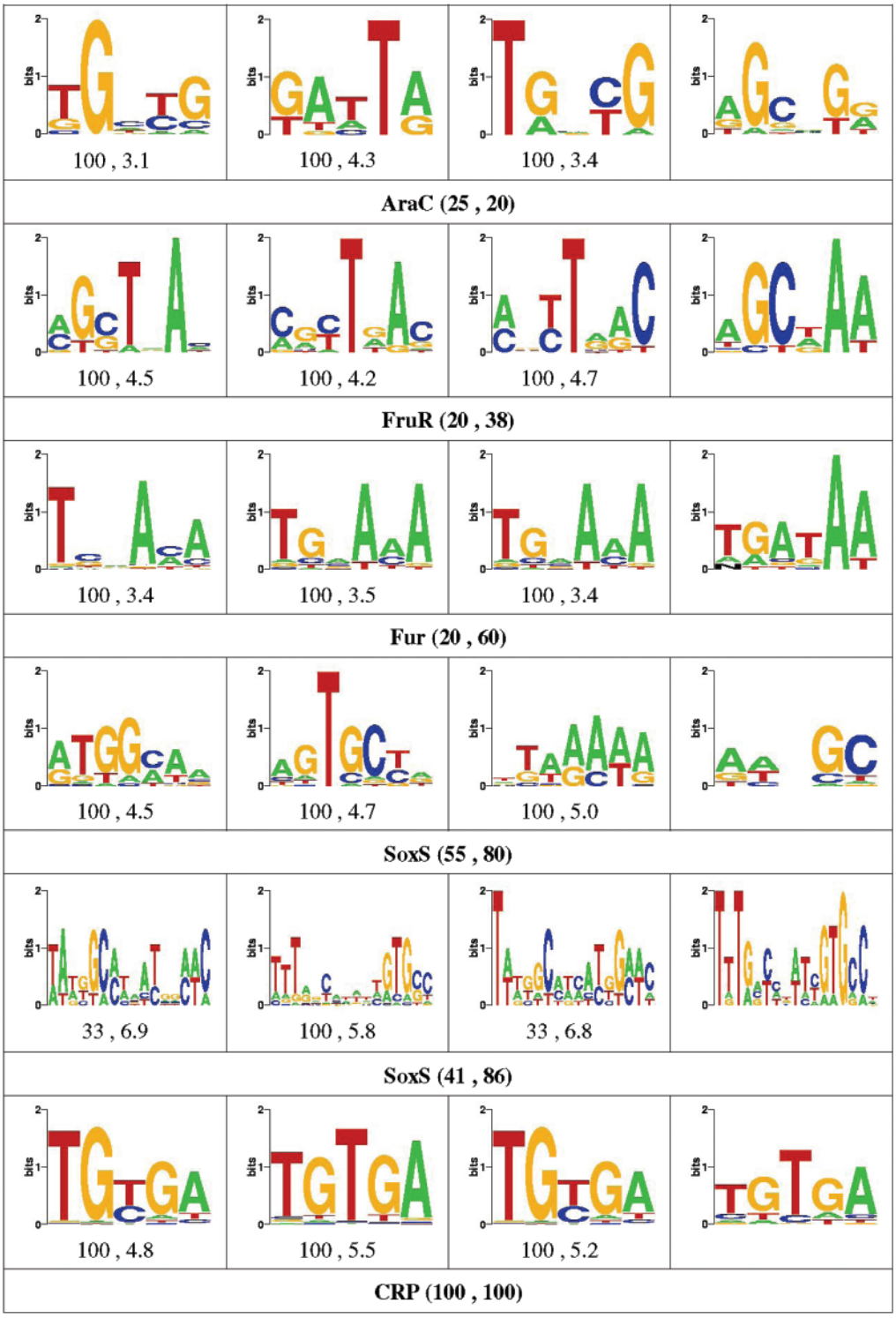


Fig. 3. Representative examples of footprint predictions using the DNASITE protocol. Binding site predictions based on comparative models for 5 *E.coli* regulators. Each row shows the results for a protein-DNA complex and the numbers in parenthesis indicate the corresponding %ID and %IID. The first three columns show the results for the P0.0001, MF and FP0.0001 parameter sets, including the % of recovered sites and the average alignment site score; the fourth shows the consensus matrix calculated by CONSENSUS/WCONSENSUS (Hertz and Stormo, 1999) on the RegulonDB sequences, as an independent control. Two independent predictions for SoxS are displayed here, using two different template complexes, one of them (55, 80) spanning only one of the DNA-contacting domains. The FP0.0001 (55, 80) prediction recovers 100% of sites, but includes false positives, as can be seen in the logo. Note that the MF (55, 80) correct prediction is also included into the (41, 86), whilst P0.0001 and FP0.0001 (41, 86) predictions do not recover all known binding sites and obtain incorrect sequence logos. SoxS is an example of split site, composed of two subsites. Our current benchmark methodology often cannot recover split sites.

Table 2. Performance of different DNASITE parameter sets tested on a total of 85 *E.coli* DNA-binding proteins with mean % sequence identity of 35 and % interface identity of 46. The first column labels each parameter set, encoded as mentioned in Materials and Methods. The second column shows the mean % of RegulonDB sites aligned with a significant score by PATSER. The third column shows the mean -ln(P) score for each DNA-binding protein, as reported by PATSER. The last column shows the mean significance of recovered sites, calculated as ln(P) – significance threshold

Parameter set	% Sites recovered	Mean –ln(P)	Mean significance
Def	94	4.7	1.5
CM	90	4.5	1.3
Sc3	94	4.6	1.7
Df1	95	4.7	1.9
Df2	94	4.6	1.5
Df3	94	4.6	1.4
C1	98	4.3	2.1
M	97	4.6	2.4
F	93	4.8	1.8
P0.01	93	4.5	1.6
P0.001	94	4.4	2.0
P0.0001	94	4.2	2.5
MF	96	4.6	2.5
FP0.001	93	4.5	2.2
FP0.0001	97	4.4	2.9

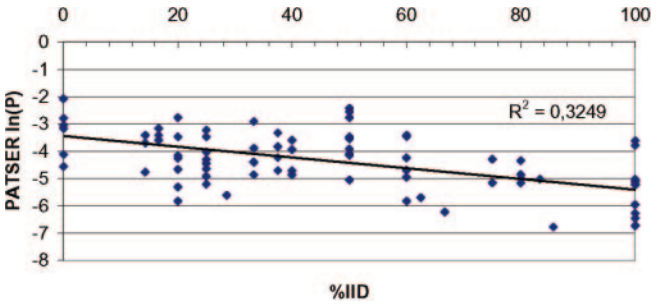


Fig. 4. Interface identity as quality predictor for DNASITE. FP0.0001 scores for 85 modelled complexes are plotted against % interface identity. The observed correlation coefficient is −0.57. This means that high IID values predict better DNASITE footprints.

mean? They suggest that keeping the conserved part of the interface from the template is a good idea (M), in agreement with previous observations (Sandelin and Wasserman, 2004), and that applying family-specific corrections helps in many cases (F). In addition, it seems to be a good choice to select only threaded sequences with low ln(P) values. The different solutions provided by each strategy might not be identical, but perhaps looking for consensus predictions may help discriminate between right and wrong predictions. 73 of these 85 predictions correspond to regulators that have more than 5 annotated binding sites in RegulonDB.

Figure 4 shows that the % interface identity (IID) correlates negatively with the obtained PATSER scores in our benchmark. The correlation coefficient ranges from −0.24 (C1) to −0.57 (FP0.0001). A linear regression line is also plotted, showing a poor R^2 value, due to the large variability of the data. A much weaker correlation is observed when % sequence identity is used instead (data not shown). This suggests that IID is really the important number when comparing different complexes, since mutations in the interface will probably mean changes in the recognised set of nucleotide sequences.

4 DISCUSSION

The assumption behind comparative modelling is that similar sequences will have very similar structures. However, similar protein structures need not have the same biological or molecular function. In our modelling problem two questions need to be answered. The first is whether a homologous protein really binds to DNA. The second is what nucleotide sequences are being recognised by this protein. We might try to answer the first question by calculating the net charge of the suspected binding protein, as suggested by Ahmad (Ahmad and Sarai, 2004), or using any related experimental evidence. However, in this work we focused on the second question.

The reported results suggest that template complexes can be used to estimate the nucleotide preferences of related proteins, as already anticipated (Morozov *et al*., 2005). These results also support the choice of FP0.0001 parameters if score significance is to be maximized. Another lesson learned here is that a conservative approach when predicting footprints is useful, keeping unchanged as much of the template complex as possible (M parameters). This could be saying that we are not very good at predicting preferred DNA sequences from scratch, perhaps because we have only tested generic recognition matrices (Pabo and Nekludova, 2000). Our results also suggest that family-specific DNA preferences can be estimated from protein sequence profiles, improving the observed alignment scores. This might help overcome the limitations of generic recognition matrices, as protein-DNA preferences might be context-specific (Kaplan *et al*., 2005). Besides family corrections, DNASITE could benefit from using tailor-made protein-DNA recognition matrices, were family-specific associations could be encoded. For instance, a homeodomain-like matrix could be derived. Preliminary work suggests that these matrices can significantly improve results but further exploration is needed.

This computational tool can generate different solutions that might be used to build a consensus. If no consensus is reached then probably the wise thing to do is to ignore these predictions. Along with the set of binding sequences selected, DNASITE also produces the motif length, a variable that non-structural footprinting methods need to estimate by other means.

DNASITE can be applied to regulators for which no experimental evidence is available at all, for instance cases where no footprint experiments have been performed. For this reason this tool can potentially be useful for the purpose of curating DNA-binding sites. Furthermore, the algorithm has been implemented using a collection of widely used tools (PSI-BLAST, SCWRL and X3DNA).

This approach makes a simplified use of interface geometry and does not explicitly distinguish H-bond interactions from Van der Waals contacts, allowing fast but perhaps less accurate predictions. Water-mediated H-bonds are also ignored as they don not seem to contribute much to specific protein-DNA recognition (Luscombe *et al*., 2001). Perhaps considering these questions would improve the method, but this remains to be tested.

A weakness of this method is that it depends on the availability of related protein-DNA complexes. For the set of approximately 300 regulators in *E.coli*, less than a third can be studied with this protocol. Probably more regulators could be modelled using more sophisticated protein alignment algorithms, but those cases would need to be benchmarked as well.

It should be remarked that a more realistic benchmark still needs to be done, using DNASITE footprints to blindly predict binding sites in the context of a genome. It is anticipated that these footprints may have relatively large false positive rates in comparison with more traditional approaches since they tend to be shorter, therefore allowing more random hits to be aligned. Therefore, future users should benefit by combining DNASITE with other structural and non-structural methods.

ACKNOWLEDGEMENTS

B.C.M. thanks Heladia Salgado for her support in using RegulonDB data, Martín Peralta for his help in constructing consensus matrices and Marc Parisien for his advice and code for calculating DNA deformation energies with X3DNA. We acknowledge suggestions by anonymous referees. This work has been supported by a post-doctoral fellowship from Universidad Nacional Autónoma de México awarded to B.C.M. and by NIH grant RO1-GM071962.

ABBREVIATIONS

PDF, Protein Data Bank; RMSD, root-median-square deviation; SCOP, structural classification of proteins; ID, sequence identity; IID, interface sequence identity; PSSM; position-specific scoring matrix.

REFERENCES

Ahmad,S. and Sarai,A. (2004) Moment-based prediction of DNA-binding proteins, *J Mol Biol*, 341, 65–71.

Aloy,P., Ceulemans,H., Stark,A. and Russell,R.B. (2003) The relationship between sequence and interaction divergence in proteins, *J Mol Biol*, 332, 989–998.

Aloy,P., Moont,G., Gabb,H.A., Querol,E., Aviles,F.X. and Sternberg,M.J. (1998) Modelling repressor proteins docking to DNA, *Proteins*, 33, 535–549.

Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res*, 25, 3389–3402.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TREMBL in 2000, *Nucleic Acids Res*, 28, 45–48.

Baker,D. and Sali,A. (2001) Protein structure prediction and structural genomics, *Science*, 294, 93–96.

Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank, *Nucleic Acids Res*, 28, 235–242.

Canutescu,A.A., Shelenkov,A.A. and Dunbrack,R.L.,Jr (2003) A graph-theory algorithm for rapid protein side-chain prediction, *Protein Sci*, 12, 2001–2014.

Contreras-Moreira,B., Ezkurdia,I., Tress,M.L. and Valencia,A. (2005) Empirical limits for template-based protein structure prediction: the CASP5 example, *FEBS Lett*, 579, 1203–1207.

Contreras-Moreira,B., Fitzjohn,P.W. and Bates,P.A. (2002) Comparative modelling: an essential methodology for protein structure prediction in the post-genomic era, *Appl Bioinformatics*, 1, 177–190.

Crooks,G.E., Hon,G., Chandonia,J.M. and Brenner,S.E. (2004) WebLogo: a sequence logo generator, *Genome Res*, 14, 1188–1190.

Chothia,C. and Lesk,A.M. (1986) The relation between the divergence of sequence and structure in proteins, *Embo J*, 5, 823–826.

Dunbrack,R.L.,Jr and Karplus,M. (1993) Backbone-dependent rotamer library for proteins. Application to side-chain prediction, *J Mol Biol*, 230, 543–574.

Gromiha,M.M., Siebers,J.G., Selvaraj,S., Kono,H. and Sarai,A. (2005) Role of inter and intramolecular interactions in protein-DNA recognition, *Gene*, 364, 108–113.

Hertz,G.Z. and Stormo,G.D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences, *Bioinformatics*, 15, 563–577.

Kaplan,T., Friedman,N. and Margalit,H. (2005) Ab initio prediction of transcription factor targets using structural knowledge, *PLoS Comput. Biol.*, 1, e1.

Kono,H. and Sarai,A. (1999) Structure-based prediction of DNA target sites by regulatory proteins, *Proteins*, 35, 114–131.

Lu,X.J. and Olson,W.K. (2003) 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures, *Nucleic Acids Res*, 31, 5108–5121.

Luscombe,N.M., Laskowski,R.A. and Thornton,J.M. (2001) Amino acid-base interactions: a three-dimensional analysis of protein-DNA interactions at an atomic level, *Nucleic Acids Res*, 29, 2860–2874.

Mandel-Gutfreund,Y., Baron,A. and Margalit,H. (2001) A structure-based approach for prediction of protein binding sites in gene upstream regions, *Pac Symp Biocomput*, 139–150.

Mandel-Gutfreund,Y. and Margalit,H. (1998) Quantitative parameters for amino acid-base interaction: implications for prediction of protein-DNA binding sites, *Nucleic Acids Res*, 26, 2306–2312.

Morozov,A.V., Havranek,J.J., Baker,D. and Siggia,E.D. (2005) Protein-DNA binding specificity predictions with structural models, *Nucleic Acids Res*, 33, 5781–5798.

Murzin,A.G., Brenner,S.E., Hubbard,T. and Chothia,C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures, *J Mol Biol*, 247, 536–540.

Nadassy,K., Wodak,S.J. and Janin,J. (1999) Structural features of protein-nucleic acid recognition sites, *Biochemistry*, 38, 1999–2017.

Olson,W.K., Gorin,A.A., Lu,X.J., Hock,L.M. and Zhurkin,V.B. (1998) DNA sequence-dependent deformability deduced from protein-DNA crystal complexes, *Proc Natl Acad Sci USA*, 95, 11163–11168.

Ortiz,A.R., Strauss,C.E. and Olmea,O. (2002) MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison, *Protein Sci*, 11, 2606–2621.

Pabo,C.O. and Nekludova,L. (2000) Geometric analysis and comparison of protein-DNA interfaces: why is there no simple code for recognition? *J Mol Biol*, 301, 597–624.

Paillard,G. and Lavery,R. (2004) Analyzing protein-DNA recognition mechanisms, *Structure* (Camb), 12, 113–122.

Raviscioni,M., Gu,P., Sattar,M., Cooney,A.J. and Lichtarge,O. (2005) Correlated evolutionary pressure at interacting transcription factors and DNA response elements can guide the rational engineering of DNA binding specificity, *J Mol Biol*, 350, 402–415.

Salgado,H., Gama-Castro,S., Peralta-Gil,M., Diaz-Peredo,E., Sanchez-Solano,F., Santos-Zavaleta,A., Martinez-Flores,I., Jimenez-Jacinto,V., Bonavides-Martinez,C., Segura-Salazar,J., Martinez-Antonio,A. and Collado-Vides,J. (2006) RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions, *Nucleic Acids Res*, 34, D394–397

Sandelin,A. and Wasserman,W.W. (2004) Constrained binding site diversity within families of transcription factors enhances pattern discovery bioinformatics, *J. Mol. Biol.*, 338, 207–215.

Sarai,A. and Kono,H. (2005) Protein-DNA recognition patterns and predictions, *Annu Rev Biophys Biomol Struct*, 34, 379–398.

Selvaraj,S., Kono,H. and Sarai,A. (2002) Specificity of protein-DNA recognition revealed by structure-based potentials: symmetric/asymmetric and cognate/non-cognate binding, *J Mol Biol*, 322, 907–915?

Siggers,T.W., Silkov,A. and Honig,B. (2005) Structural alignment of protein--DNA interfaces: insights into the determinants of binding specificity, *J Mol Biol*, 345, 1027–1045

Steffen,N.R., Murphy,S.D., Tolleri,L., Hatfield,G.W. and Lathrop,R.H. (2002) DNA sequence and structure: direct and indirect recognition in protein-DNA binding, *Bioinformatics*, 18 (Suppl 1), S22–30.

Tramontano,A., Leplae,R. and Morea,V. (2001) Analysis and assessment of comparative modeling predictions in CASP4, *Proteins* (Suppl), 22–38.

Wang,G. and Dunbrack,R.L.,Jr. (2003) PISCES: a protein sequence culling server, *Bioinformatics*, 19, 1589–1591.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e81–e89
doi:10.1093/bioinformatics/btl252

A probabilistic approach to protein backbone tracing in electron density maps

Frank DiMaio[1,2,*], Jude Shavlik[1,2] and George N. Phillips[3,1]

[1]Computer Sciences Dept., [2]Biostatistics and Medical Informatics Dept. and [3]Biochemistry Dept., University of Wisconsin—Madison, USA

ABSTRACT

One particularly time-consuming step in protein crystallography is interpreting the electron density map; that is, fitting a complete molecular model of the protein into a 3D image of the protein produced by the crystallographic process. In poor-quality electron density maps, the interpretation may require a significant amount of a crystallographer's time. Our work investigates automating the time-consuming initial backbone trace in poor-quality density maps. We describe ACMI (Automatic Crystallographic Map Interpreter), which uses a probabilistic model known as a Markov field to represent the protein. Residues of the protein are modeled as nodes in a graph, while edges model pairwise structural interactions. Modeling the protein in this manner allows the model to be flexible, considering an almost infinite number of possible conformations, while rejecting any that are physically impossible. Using an efficient algorithm for approximate inference—belief propagation—allows the most probable trace of the protein's backbone through the density map to be determined. We test ACMI on a set of ten protein density maps (at 2.5 to 4.0 Å resolution), and compare our results to alternative approaches. At these resolutions, ACMI offers a more accurate backbone trace than current approaches.

Contact: dimaio@cs.wisc.edu

1 INTRODUCTION

Determining the folding of a protein—that is, the three-dimensional spatial configuration of the atoms in a protein—has long been an important problem in biochemistry. With some exceptions, a protein's structure is uniquely determined from its linear amino-acid sequence. Unfortunately, no known algorithm can determine this unique structure from sequence, and scientists are forced to rely upon laboratory methods in order to determine protein structures. Several experimental methods exist, the most popular of which—accounting for about 80% of protein structures determined to date—is x-ray crystallography.

There has been significant recent interest in high-throughput structure determination [1]. One particularly time-consuming step in crystallography is interpretation of the electron map, that is, finding the location of all the protein's atoms in a three-dimensional image of the protein. In this paper, we describe ACMI (Automatic Crystallographic Map Interpreter), an algorithm that automates the process of tracing the backbone in electron density maps.

ACMI consists of two main components: a *local matching* component that locates individual amino acids in the density map, and a *global constraint* component that uses prior knowledge of the protein's structure to eliminating false positives from the local matching. ACMI combines these two with an efficient inference algorithm that can infer the protein's backbone in an electron density map. ACMI's model is *probabilistic*: throughout the interpretation it represents each residue as a probability distribution over the electron density map. This property—not being contrained to force each residue to a single location—is advantageous as it naturally handles noise in the map, errors in the input sequence, and disordered regions in the protein.

2 CRYSTALLOGRAPHY BACKGROUND

Protein crystallography is a very labor-intensive undertaking. First, the protein must be produced in large quantities and purified. Protein crystals then have to be grown, which usually requires testing a significant number of crystallization conditions and solvents. Once the crystals are finally available, a beam of x-rays is fired through the crystal. The lattice of protein molecules that comprise the crystal diffracts this x-ray beam, and produces a pattern of spots on a plate. These spots represent the intensities of a Fourier-transformed picture of the protein. Further laboratory experiments are used to determine the phases corresponding to these intensities. Finally, a Fourier transform converts these intensities into an *electron density map*: a three-dimensional image of the protein.

The final step in x-ray crystallography is *interpreting* this electron density map, converting it into a representation that is usable by biologists. During interpretation, the crystallographer must locate—given the amino-acid sequence of the protein—the coordinates of the centers of all the protein's atoms. This interpretation can be extremely time-intensive; a crystallographer may spend weeks (even months!) interpreting a poor-quality electron density map.

The electron density map is defined on a 3D lattice of points covering the *unit cell*, or basic repeating unit in the protein crystal. The crystal's unit cell may contain multiple copies of the protein related by *crystallographic symmetry*, one of the 65 regular ways a protein can pack into the unit cell. Rotation/translation operators relate one region in the unit cell (the *asymmetric unit*) to all other symmetric copies. Furthermore, the protein may form a multimeric complex (e.g. a dimer, tetramer, etc.) within the *asymmetric unit*. In all these cases is up to the crystallographer to isolate and interpret a single copy of the protein.

An overview of the interpretation task is illustrated in Figures 1 and 2. In both figures, the electron density map—a 3D function over the unit cell—is illustrated as an isocontoured surface. Figure 1a

*To whom correspondence should be addressed.

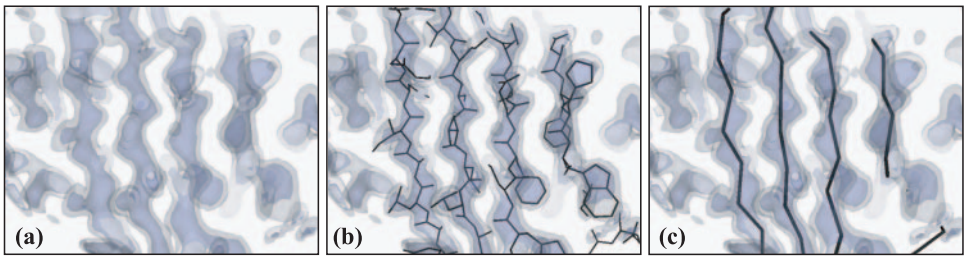


Fig. 1. An over view of electron density map interpretation. Given the amino acid sequence of the protein and a density map (a), the crystallographer's goal is to find the positions of all the proteins atoms (b). Alternatively, a backbone trace (c), reduces each residue to a single point. ACMI automates determination of the backbone trace.

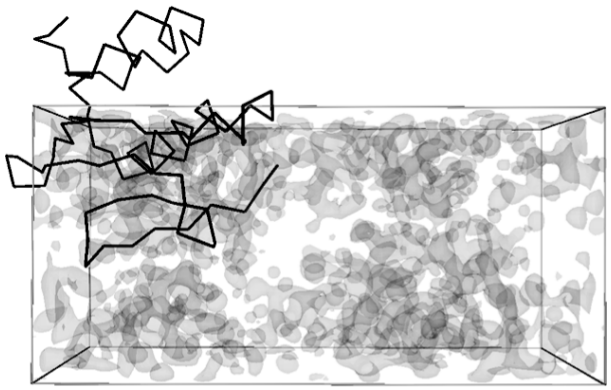


Fig. 2. The electron density map over an entire unit cell. One copy of the protein is indicated. This unit cell contains two symmetric copies, which wrap around the map boundary.

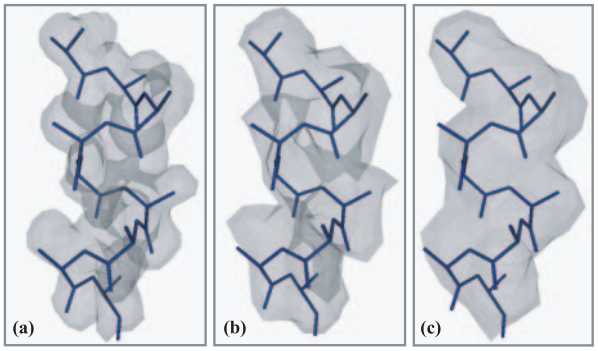


Fig. 3. The electron density map for the same protein fragment at (a) 2Å, (b) 3Å, and (c) 4Å map resolution.

illustrates a small portion of the electron density map. The sticks in Figure 1b show the location of bonds between atoms. Figure 1c shows only the lines between adjacent C_α atoms of the protein. This C_α trace (or backbone trace) is the main concern of this paper. Finally, Figure 2 shows the scale of the problem, illustrating a complete unit cell's electron density. This map contains two crystallographically symmetric copies of a protein.

One measure of overall quality of an electron density map is the *resolution* of the map. When placed in an x-ray beam, some protein crystals diffract the beam better than others. In general, the more the crystal diffracts the beam, the better quality the map. This is illustrated in Figure 3, which shows a short protein's electron density at a variety of resolutions (lower values of resolution mean a higher-quality electron density map). At high resolutions (2Å or better resolution) individual atoms are visible, and automated interpretation is straightforward [2]. However, above about 2.5Å, details of individual atoms are smeared, and atom-based methods tend to fail. Several approaches have attempted automatically interpreting these maps [3,4] and have met with some success. However, interpretations produced by these methods are often messy and require significant crystallographer effort to ''fill in the gaps.''

3 OVERVIEW OF THE ALGORITHM

A high-level overview of ACMI's two main components is illustrated in Table 1. ACMI includes a *local matching* component, where individual residues are probabilistically located in the map, independent of all other residues, and a *global constraint* component, where the backbone chain is built up, also probabilistically, from the local matches, taking into account the chemical laws governing the physical structure of proteins.

The local-matching component of our algorithm makes use of a library of existing sequence-specific 5-mer templates. That is, when searching for an individual residue, we actually look for all common conformations of the 5-mer centered at that residue. The local search has high sensitivity, usually matching well to the residue's correct location. However, it suffers from low specificity, producing a significant number of false positives.

ACMI's global-constraint component probabilistically refines these local search results to takes into account prior knowledge of protein structure. Using this prior knowledge, it adjusts the local-match probabilities based on the local match probabilities of other residues. It produces a *physically feasible* interpretation that maximizes the probabilities from the local matching.

ACMI models this physical feasibility with a pairwise Markov field, which represents the probability of a conformation as the product of probabilities between pairs of residues. This pairwise potential is analogous to the pairwise potential energy calculations used in molecular dynamics [5] (although our model does not optimize physical energy but rather *statistical* ''energy'').

4 LOCAL MATCHING

Local matching in ACMI is used to locate individual protein residues in an electron density map. In the poor-quality maps for which ACMI is designed, simple atom-based-refinement methods [2] perform poorly. Empirically, methods using rotamer searching [6], skeletonization [7], or critical points [8] also perform poorly in

Table 1. A pseudocode overview of ACMI's algorithm

Procedure ACMI

Given: *sequence* 'seq' and electron density map '**M**'
Find: *putative backbone trace* **W**={ w_i }
  foreach residue *i* do
    $P(\boldsymbol{M}|w_i) \leftarrow$ **doLocalMatch**(*seqi*, **M**)
  $P(\boldsymbol{W}) \leftarrow$ **enforceGlobalConstraints**($P(\boldsymbol{M}|w_i)$)
  optimal_trace $\leftarrow \{w_i^* | \forall i\ w_i^* = \text{argmax}\ (P(w_i))$

Procedure doLocalMatch(*seq, M*)
Given: *sequence* 'seq' and electron density map '**M**'
Find: *prob. dist.* P(**M**|w_i) *of each residue over map*
- Consider 5-mer centered at each residue
- Extract instances of 5-mer from PDB, cluster to characterize 5-mer's conformational space
- Perform a 6D search for 5-mer over density map
- Use a tuning set to convert squared density differences to probabilities $P(\boldsymbol{M}|w_i)$ for each residue *i*

Procedure enforceGlobalConstraints($P(\boldsymbol{M}|w_i)$)
Given: individual residue *probability distributions*
Find: marginal *probabilities given structure constraints*
- Model protein backbone structure as a graph
  - *Nodes* model α-carbon positions
  - *Edges* enforce structural constraints
- Probability of an interpretation $\mathbf{W} = \{w_i\}$ given as the product of node potentials and edge potentials

$$P(\mathbf{W} \mid \mathbf{M}) \propto \prod_{\text{residues } i,j} P(w_i, w_j) \prod_{\text{residues } i} P(\mathbf{M} \mid w_i)$$

- Infer marginal probs. given structural constraints

these low-resolution maps. The methods that have had the most success in low-resolution maps are those based upon finding large fragments of protein electron density [9]. Thus, we use *sequence-specific* 5*-mer search* to locate individual residues in the electron density map.

Our method is divided into two basic parts, illustrated in Figures 4 and 5. First, we use previously solved structures from the Protein Data Bank to construct a basis set of sequence-specific 5-mer templates. We then perform a 6D (rotation + translation) search in the map for each of the 5-mers in our basis set. The output of this local search is—for each residue—an estimated probability distribution of that residue's presence over the unit cell.

Constructing a Sequence-Specific 5*-mer Basis Set.* ACMI begins this step—illustrated in Figure 4—by walking along the one-dimensional protein sequence, considering a 5-mer centered at each residue. Given this 5-mer, we search a *non-redundant subset* of the PDB [10] (restricted to have less than 25% sequence similarity) for three-dimensional instances of that 5-mer. If there are less than 50 such instances then we search for near neighbors to the 5-mer using increasing PAM distance [11] until we have 50 structures.

It is infeasible to search for all 50+ conformations in the electron density map, so we instead cluster the structures and represent each cluster as a centroid fragment and a weight. When clustering the fragments, we use rotationally-aligned all-atom RMS deviation between fragments as a distance metric (quickly computed as an optimization problem [12]). We use complete-linkage hierarchical

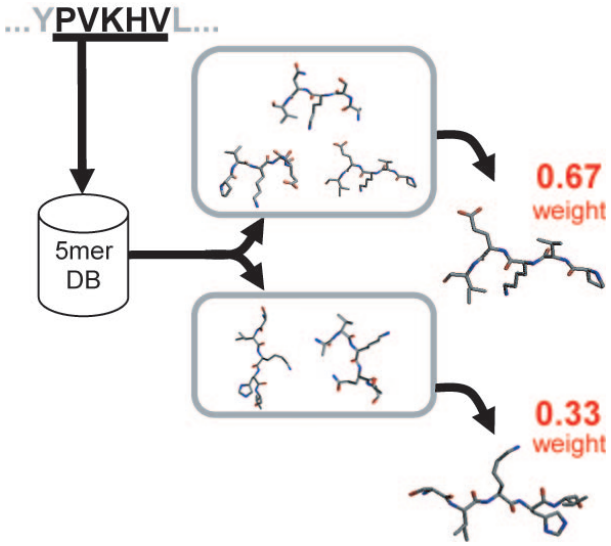


Fig. 4. The 5-mer clustering process. Walking along the amino-acid sequence, we consider a 5-mer centered at each position. We search the database for instances of that 5-mer, and cluster them. Finally, we extract a representative member from each cluster. This characterizes the conformational space of the 5-mer sequence.

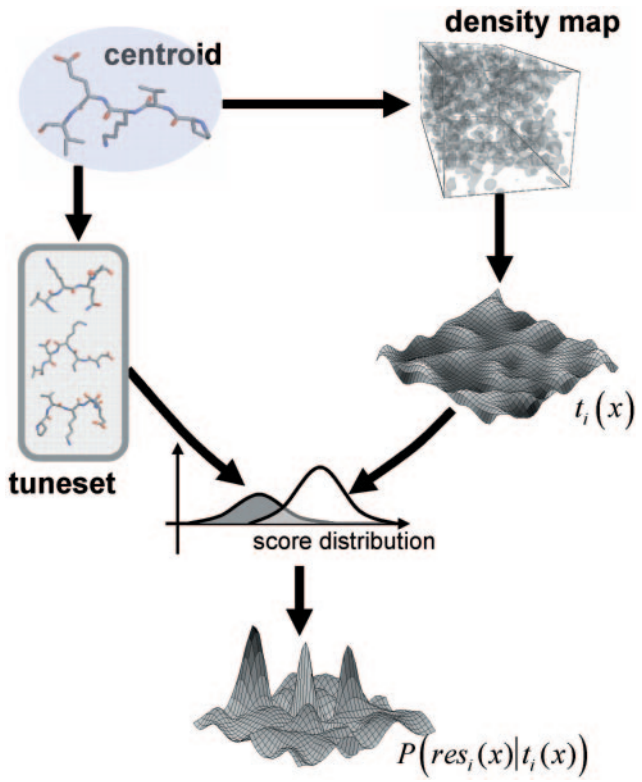


Fig. 5. An overview of the 5-mer template matching process. After we have extracted a representative et of 5-mers for each residue *i*, we perform a 6D (rotation + translation) search for the fragment in the density map. By also matching the fragment to a tuning set of known structures, we can use Bayes' rule (see Equation 3) to determine the probability distribution of the residue over the density map.

clustering, limiting clusters to have a maximum diameter of 3Å. Any cluster with under 10% representation is thrown out (to limit CPU time in the next step); in all remaining clusters we find a centroid (i.e. representative) fragment. We also record the cluster weight with each centroid fragment, that is, the percent of structures that fell into that cluster. Depending on the ''sequence structural entropy'' of the 5-mer [13], anywhere from 1 to 7 clusters (and resultant centroid fragments) are produced.

The cluster centroids and the weights determined by ACMI represent the conformational space of each specific 5-mer fragment. Using fragments of length five is our way of balancing the trade-off between template size and template specificity. Larger fragments are preferred for recognition in poor-quality maps, but larger fragments have lower representation in the set of already-solved structures. Our non-redundant PDB subset contains about 20% of the 3.2×10^6 possible 5-mers.

Searching for 5-mer centroid fragments. Once the clustering is complete and the cluster centroids have been extracted, we search for instances of the centroids in the electron density map. This process is illustrated in Figure 5. Given a fragment and a target resolution, we can build a map corresponding to what we would expect to see, given the fragment. Then, at each map location, we can compute the mean squared electron density difference $t(\vec{x})$ between the map and the fragment. We compute this difference over all points $\vec{x} = < x_i, y_i, z_i >$ in the electron density map within some distance of the fragment,

$$t(\vec{x}) = \sum_y \varepsilon_f(\vec{y}) \left(\rho'_f(\vec{y}) - \frac{1}{\sigma_\rho(\vec{x})} [\rho(\vec{y}-\vec{x}) - \bar{\rho}(\vec{x})] \right)^2 \qquad (1)$$

where $\rho(\vec{x})$ is the map in which we are searching, $\rho'_f(\vec{x})$ is the standardized fragment electron density, $\varepsilon(\vec{x})$ is a masking function that is nonzero only for points near the fragment, and $\sigma_\rho(\vec{x})$ scales the standard deviations of the fragment and map densities,

$$\sigma^2_\rho(\vec{x}) = \sum_y \varepsilon_f(\vec{y}) [\rho(\vec{y}-\vec{x}) - \bar{\rho}(\vec{y})]^2 \Big/ \sum_y \varepsilon_f(\vec{y}) \qquad (2)$$

We need to perform the fragment search as a 6D search over all rotations plus all translations; fortunately, we can compute $t(\vec{x})$ quickly at a single rotation using FFTs [14]. Additionally, at each position we store the best-matching 5-mer fragment, and the corresponding rotation, for later use.

The electron density difference function $t(\vec{x})$ is a good measure of similarity between regions of density, but we need a way to convert these scores into probability distributions, that is, the probability $P(\vec{x}_i \mid score_i)$ that there is an instance of a specific 5-mer cluster i at location $\vec{x}_i$ given match score $score_i$. ACMI computes this using a tuning set and the application of Bayes' rule. Bayes' rule states that this probability is given as

$$P(\vec{x}_i \mid score_i) = P(score_i \mid \vec{x}_i) \cdot \frac{P(\vec{x}_i)}{P(score_i)} \qquad (3)$$

The terms on the right-hand side are computed or estimated as follows. The probability distribution of match scores over the map, $P(score_i)$, is derived from the actual distribution of match scores over the (unsolved) map. The prior probability on a residue's location over the map, $P(\vec{x}_i)$, is simply a normalization term: we already know (by knowing the protein's sequence) the number of copies of the 5-mer in the electron density map, and we normalize probabilities over the map to sum to this value. However, the first term—the distribution of scores when the 5-mer *matches* the map—is trickier to compute. ACMI *estimates* this term using a tuning set derived from different protein structures from the PDB. This tuning set contains *other instances* from the 5-mer cluster for which we are searching. We match each cluster centroid's density map with each tuneset density map in that centroid's cluster to estimate the distribution of scores given a 5-mer match.

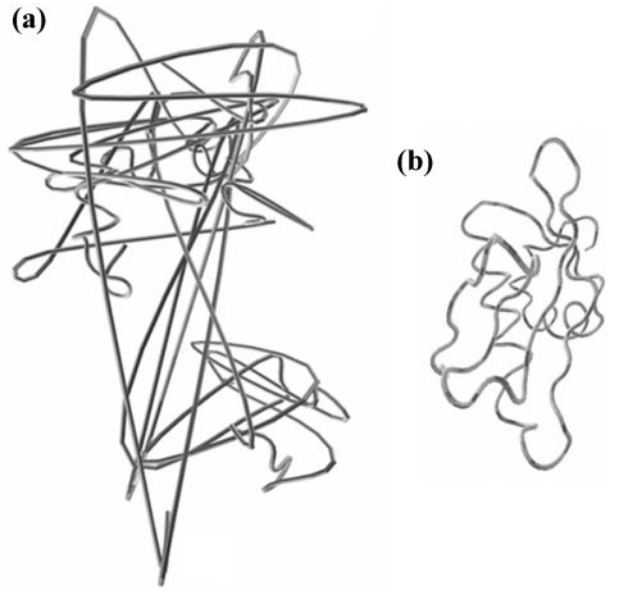


Fig. 6. Two possible backbone traces. The trace (a) maximizes the product of 5-mer match probabilities; however, the resultant protein is physically impossible. We would prefer trace (b) with a lower 5-mer match probability, but which corresponds to a physically-possible structure.

At the end of the local matching procedure, ACMI has computed—for each residue—a probability distribution over the unit cell. That is, for each point in 3D space, we have a probability that each specific 5-mer is positioned at that location. The remainder of the paper describes how our algorithm uses prior knowledge about the structure of the protein to estimate the most probable backbone trace given these probability distributions. Run times for the local matching are significant: for each fragment we have to search ~1900 rotations (20-degree discretization) over the entire electron density map. The total compute time is on the order of CPU-weeks; however, 5-mer matching is trivially parallelized [15].

5 GLOBAL CONSTRAINTS

In Section 4, we computed—for each residue i—the probability distribution over every position x in the unit cell. We can think of this probability as the probability that this map was *generated* by residue i at location and rotation $\vec{w}_i$, that is, $P(\mathbf{M} \mid \vec{w}_i)$. One could presumably select, for each residue, the $\vec{w}_i$ that maximized this probability. However, the resultant trace would likely look like that in Figure 6a. ACMI somehow needs to account for the *structural probability* on the model. That is, it needs to ensure that the proposed structure is a *physically feasible* protein molecule. What we ultimately want to find—given map **M**—is the configuration of

all residues $\mathbf{W} = \{\vec{w}_1, \ldots, \vec{w}_N\}$, such that

$$\arg\max_{\mathbf{W}=\{w_1,\ldots,w_N\}} P(\mathbf{W}\,|\,\mathbf{M}) \propto P(\mathbf{W}) \cdot \prod_{i=1\ldots N} P(\mathbf{M}\,|\,\vec{w}_i) \tag{4}$$

The first term accounts for this physical feasibility, in which a proposed structure like that of Figure 6b would have a much higher probability of configuration than Figure 6a.

5.1 Markov field model

To model this "global constraint" probability ACMI uses a *pairwise Markov field model* [16]. A pairwise Markov field model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $i \in \mathcal{V}$ connected by edges $(i,j) \in \mathcal{E}$. Each node in the graph is associated with a (hidden) random variable $\vec{w}_i \in \mathbf{W}$. The graph is conditioned on observation variables $\mathbf{M}$. Each vertex has a corresponding *observation potential* $\psi_i(\vec{w}_i, \mathbf{M})$, and each edge is associated with a *conformational potential* $\psi_{ij}(\vec{w}_i, \vec{w}_j)$. We can represent the full joint probability as

$$p(\mathbf{W}\,|\,\mathbf{M}) = \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(\vec{w}_i, \vec{w}_j) \cdot \prod_{i\in\mathcal{V}} \psi_i(\vec{w}_i, \mathbf{M}) \tag{5}$$

We are concerned with finding the $\vec{w}_i \in \mathbf{W}$ maximizing this probability, given some $\mathbf{M}$.

Figure 7 shows how we encode a protein in a Markov field model. Each node i represents an amino-acid residue in the protein. The label $\vec{w}_i = \langle \vec{x}_i, \vec{q}_i \rangle$ for each amino-acid residue consists of seven terms: the 3D Cartesian coordinates $\vec{x}_i$ of the residue's *alpha Carbon* (C_α), and four internal parameters $\vec{q}_i$ (an alternate parameterization of three rotational parameters plus the ''bend'' angle formed by three consecutive residues). The *observation potential* $\psi_i(\vec{w}_i, \mathbf{y})$ associated with each residue is the 5-mer probability $P(\mathbf{M}\,|\,\vec{w}_i)$ computed in Section 4.

The *conformation potentials* $\psi_{ij}(\vec{w}_i, \vec{w}_j)$, which model the probability of a particular conformation of the residues in the protein, are further divided into two basic types. Following Suddereth *et al.*'s hand-tracking model [17], ACMI defines *adjacency potentials* associated with each edge connecting neighboring residues (Figure 7b). These potentials ensure that adjacent residues maintain the proper 3.8Å spacing and the proper C_α—C_α—C_α angle. ACMI also defines *occupancy potentials* between non-adjacent residues (Figure 7c), which prevent two residues from occupying the same region in three-dimensional space. Thus, our joint probability is now defined

$$p(\mathbf{W}, \mathbf{M}) = \prod_{\substack{\vec{w}_i,\vec{w}_j\in\mathbf{W} \\ i,j \text{ adjacent}}} \psi_{adj}(\vec{w}_i, \vec{w}_j) \times \prod_{\substack{\vec{w}_i,\vec{w}_j\in\mathbf{W} \\ i,j \text{ nonadjacent}}} \psi_{occ}(\vec{w}_i, \vec{w}_j) \times \prod_{w_i\in\mathbf{W}} P(\mathbf{M}\,|\,\vec{w}_i) \tag{6}$$

Because residues distant on the protein chain are not necessarily distant in space, the graph must be fully connected; that is, every pair of residues is joined by an edge in the Markov field model.

5.1.1 Adjacency potentials The adjacency potentials, which connect every adjacent pair of residues, are further broken down into the product of two constraining functions, a distance constraint function and a rotational constraint function

$$\psi_{adj}(\vec{w}_i, \vec{w}_j) = p_x(\|\vec{x}_i - \vec{x}_j\|) \cdot p_\theta(\vec{w}_i, \vec{w}_j) \tag{7}$$

The distance constraint is based on the physical fact that, in proteins, the C_α—C_α distance is a nearly invariant 3.8Å. Thus, the potential p_x takes the form of a tight Gaussian around this ideal value.

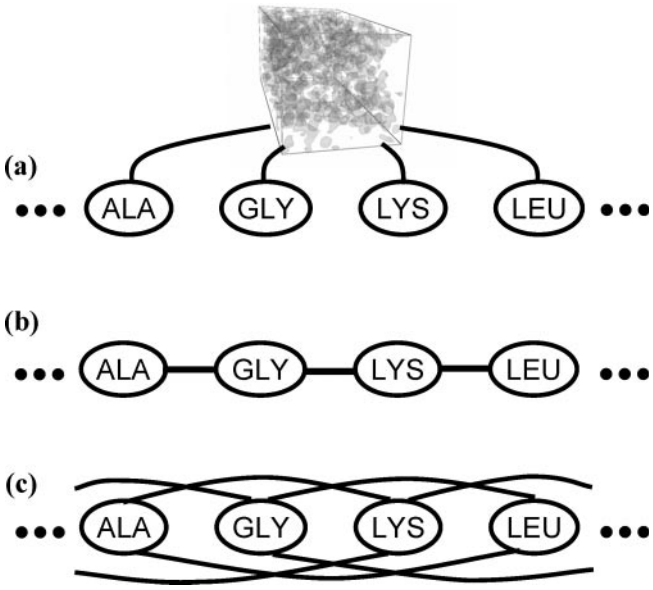


Fig. 7. The structure of our graphical model. The joint probability of a conformation of residues is the product of (a) an observation potential at each node, (b) an adjacency potential between adjacent residues, and (c) an occupancy potential between all pairs of non-adjacent residues.

The internal parameters $\vec{q}_i$ model the 3D rotation of each residue and the angle formed by the residue triple centered at residue i. To simplify the definition of p_θ , we choose to parameterize these four degrees of freedom as two pairs of θ-φ spherical coordinates: the most likely direction of the forward $(i+1)$ residue and the backward $(i-1)$ residue. Our local 5-mer matching of Section 4—in addition to computing the probability at a specific location—also remembers the most likely 5-mer centroid and rotation of that centroid. *At each location in the map*, we store four values—θ_f, φ_f, θ_b, and φ_b—indicating the direction of both adjacent residues, based on the direction of these residues in this rotated, best-matching 5-mer.

The angular constraint function p_θ, illustrated in Figure 8, is then—at each position x_i in the map—just a fixed-width Gaussian on a sphere, centered on this preferred orientation. That is, given residue i at the center of the sphere, the highest potential p_θ is when residue i+1 is located on the lightest points on the sphere, at $\langle \theta_f, \varphi_f \rangle$.

5.1.2 Occupancy potentials Occupancy potentials are in place to ensure that two residues do not occupy the same location in space. They are defined independently of orientation, and are merely a step function that constrains two (nonadjacent) C_α's be at least 3.0Å apart (the closest distance two nonadjacent residues may get),

$$\psi_{occ}(w_i, w_j) = \begin{cases} 1 & \|x_i - x_j\| \geq 3.0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

It is in this structural potential function that ACMI deals with crystallographic symmetry. We can slightly modify our potential function so that—given symmetric operators $K = \{K_n\}$—two residues may not occupy the same space, nor may *any of their*

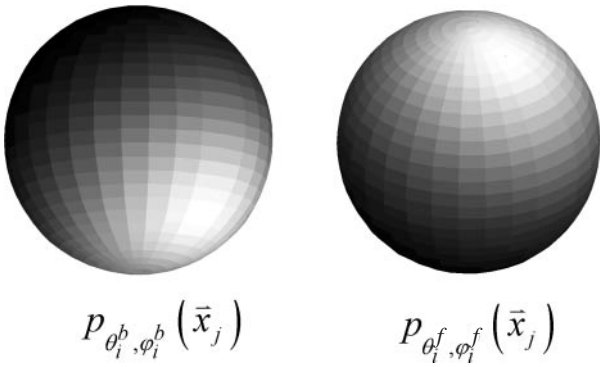


Fig. 8. The angular component $p_\theta(w_i, w_j)$ of ACMI's adjacency potential. When performing our 5-mer matching, ACMI remembers the positions of the adjacent residues in the most-likely match. The potential p_θ is a Gaussian on the sphere's surface centered on this most likely location of each adjacent residue. *This figure shows p_θ at a single location x_i in the unit cell.*

symmetric copies:

$$\psi_{occ}(w_i, w_j) = \begin{cases} 1 & \left(\min_{\substack{\text{symmetric} \\ \text{transforms } K}} \|x_i - K_n(x_j)\| \right) \geq 3.0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Multiple chains in the asymmetric unit are also handled by ACMI: separate chains are fully connected by edges enforcing occupancy constraints.

5.2 ACMI's inference algorithm: finding the most probable backbone trace

The ultimate goal of ACMI is producing a backbone trace: finding the labels $\mathbf{W} = \{w_i\}$ that maximize the probability of the local *observational potentials* and the global *conformational potentials*,

$$\arg\max_{\mathbf{W}=\{w_i\}} \prod_{\text{residues } i,j} \psi_{ij}(w_i, w_j) \cdot \prod_{\text{residue } i} \psi_i(w_i, \mathbf{M}) \quad (10)$$

However, solving this exactly for arbitrary graphs is infeasible (dynamic programming can solve this in quadratic time for tree-structured graphs). As an alternative, ACMI uses belief propagation (BP) to compute an approximation to the marginal probability $P(w_i \mid \mathbf{M})$ for each residue i, then chooses the maximum marginal label for each residue as the final trace.

Belief propagation is an inference algorithm—based on Pearl's polytree algorithm [18]—that computes marginal probabilities using a series of local messages. At each iteration, a node (i.e., residue) computes an estimate of its marginal distribution (i.e., an estimate of the residue's location in the unit cell) as the product of all incoming messages. The residue then passes a convolution of this product with the corresponding edge potential along each outgoing edge.

$$m_{i\to j}^n(w_j) \propto \int_{\text{unitcell}} \psi_{ij}(w_i, w_j) \cdot \frac{\hat{p}_i^n(w_i)}{m_{j\to i}^{n-1}(w_i)} dw_i \quad (11)$$

Above, $\hat{p}_i^n$ denotes the estimation of i's marginal at iteration n, that is,

$$\hat{p}_i^n(w_i) \propto \psi_i(w_i, \mathbf{M}) \cdot \prod_{k\in\Gamma(i)} m_{k\to i}^n(w_i) \quad (12)$$

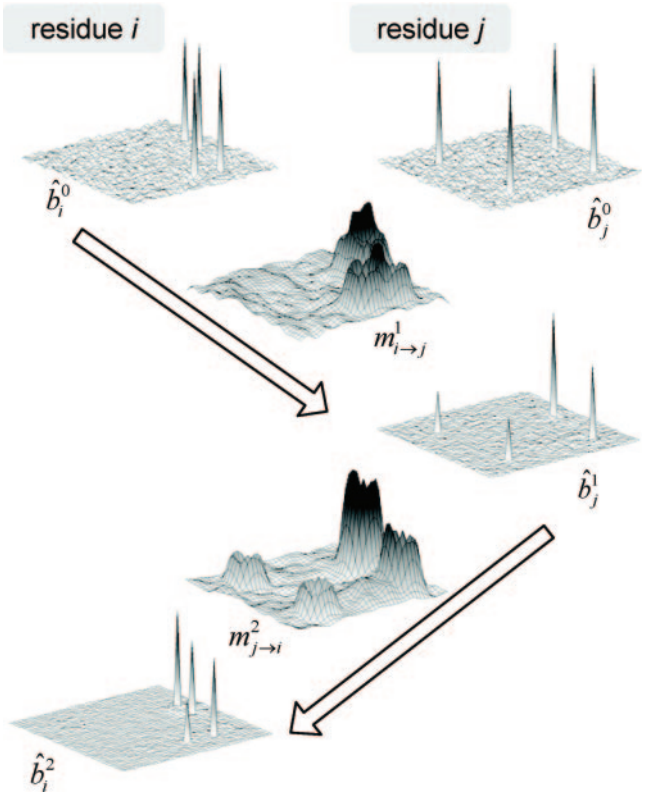


Fig. 9. A simple example of message passing using belief propagation. Given prior probabilities $\hat{b}_i^0$ and $\hat{b}_j^0$, at each iteration, a node i passes a message to a node j indicating i's belief of j's position. For example, a residue knows that an adjacent residue must be 3.8Å away; residue i's message to j consists of these 3.8 Å "bubbles" around its peaks. As BP iterates, the matches that are *structurally supported* by other residues begin to emerge.

Figure 9 illustrates the message-passing with a simple two-dimensional example. In this example, two residues' prior probabilities have their probability mass split among several peaks. Our structural knowledge tells us that residue i must be next to residue j. In the first iteration, residue i passes a message to residue j, that indicates where residue i expects to find residue j (essentially, in a ring around residue i's peaks). Messages in BP are probability distributions marginalized to the message recipient's random variables; that is, this message from residue i to residue j is a function over residue j's position in the unit cell. Residue j passes a message back to residue i indicating where j expects to find i. This example shows that in just two iterations, BP is able to reduce the number of peaks through the use of structural priors.

In graphs without cycles, BP is exact. In graphs with arbitrary topologies, such as ACMI's protein model, there are no guarantees of convergence or correctness; however, empirical results show that *loopy BP* often produces a good approximation to the true marginal [19,20].

5.3 Technical challenges

Even with the computational savings afforded by BP, the size and complexity of both the graph and the space of labels presented ACMI with a number of implementation challenges. Though

beyond the scope of this paper, the modifications necessary to BP in order to scale to this problem are discussed in another paper by the authors [21]. This section will briefly discuss some of these scaling issues.

5.3.1 Representation of potentials The label associated with each residue is a continuously-valued, 7-dimensional variable. Nonparametric belief propagation (NBP) [17] is a variant of BP that can handle continuous-valued labels; previous work represented the belief as the sum-of-Gaussians. Our work introduces Fourier-Series NBP, a variant of NBP which represents messages and belief as a set of 3D Fourier coefficients in Cartesian space, which offer a number of benefits for this problem domain. These benefits include natural treatment of periodic boundary conditions and symmetry, no explicit initialization required (as is required with the sum-of-Gaussians), and an efficient message-passing implementation.

5.3.2 Efficient message passing Each message passed requires integrating over the entire unit cell, which naïvely takes running time of the order $O(K^2)$, where K is the number of Fourier coefficients. Unfortunately, for a typical protein, K may be 10^6 to 10^7! For adjacency messages, it is not too much of a problem, as we only need to integrate over a thin spherical shell where ψ_{adj} is nonzero. However, for occupancy messages, this message computation time is significant. Fortunately, because the occupancy potential is only a function of the *distance* between the two connected residues, we can pass the message in $O(K \log K)$ as a multiplication in Fourier-space.

5.3.3 Structural message aggregation Because our graph is fully connected, in each iteration $O(N^2)$ messages need to be computed and stored, where N is the number of amino-acid residues in the protein. As each message is a probability distribution over the entire unit cell, this is demanding computationally and storage-wise. However, the outgoing structural messages (see Equation 11) at a given node are all quite similar: they only differ in the denominator, which serves to avoid double-counting, making the method exact in tree-structured graphs [19]. However, in loopy graphs, this double-counting is unavoidable. Furthermore, the structural potentials are very diffuse, high-entropy potentials. Other authors have suggested [22] that approximation errors in graphs with this type of potential tend to stabilize.

We can save a significant amount of work if we aggregate all the non-bonded residues, sending them a single structural message (that is, dropping the denominator). ACMI does this, only sending $O(N)$ messages per iteration. Combined, these BP optimizations allow ACMI to handle large proteins with large unit cells. Typical run times (for the BP inference) vary from several hours to a day.

6 EXPERIMENTS

We obtained a set of ten model-phased electron density maps from the Center for Eukaryotic Genomics at the University of Wisconsin-Madison. The maps are all of fairly good resolution—natively 1.5 to 2.5Å—and all have crystallographer-determined solutions. To test ACMI's performance on poor-quality (2.5+ Å) data, we down-sampled these maps by *smoothly diminishing* the intensities of higher-resolution reflections. To avoid truncation effects, and give a more realistic model of low-resolution data, we scaled structure factors by $\exp(-K/R^2)$, where R is the resolution of the structure factor and K is a scaling constant chosen based on the desired resolution (higher values of K smooth the map more). We down-sampled each of our maps to 2.5, 3.0, 3.5, and 4.0 Å resolutions, giving us a total of 32 maps on which to test. We chose $K = R_0^2$, so the signal strength was weakened by $1/e$ at the point of truncation.

We compared the performance of ACMI on these maps to two other automated techniques specialized to low-resolution maps: Ioerger's TEXTAL [4], and Terwilliger's Resolve [3,6]. These two approaches have had the most success handling interpretation in poor-quality maps.

TEXTAL is based on ideas from pattern recognition. Ioerger constructs a set of 15 rotation-invariant density features. Using these features at several radii, TEXTAL trains a neural network to identify C_α atoms. Sidechains are identified by looking at the electron density around each putative alpha carbon, efficiently finding the most similar region in a database, and laying down the corresponding sidechain.

Terwilliger takes a different approach with Resolve. Resolve first looks for large secondary-structure elements, places them into the map, and extends them. A rotamer search places sidechains, aligning sequence to backbone. Both methods have some success in 2.5 to 3.5 Å maps.

After running all three algorithms on the test set, we measured the results using three different metrics:

(1) C_α RMS error between predicted and true structure

(2) percent of the chain solved

(3) percent correct residue identity

Ideally, a method would find a trace with low RMS error, high percent of the chain solved, and high residue identity.

The results at each resolution are summarized in Figure 10. TEXTAL was unable to run on one protein's density maps (at any resolution)—rather than including a terrible score for this map, we gave the benefit of the doubt to TEXTAL and only report results on the nine maps on which it ran. In terms of RMS error (Figure 10a), our algorithm consistently out performs TEXTAL at all resolutions tested. Using a two-tailed pair t test, ACMI outperforms TEXTAL with p values of 0.091, 0.057, 0.012 and 0.11 at 2.5, 3, 3.5, and 4Å, respectively. Resolve performs roughly equivalent to ACMI at 2.5Å resolution; however, at 3, 3.5 and 4 Å, ACMI's performance is much better: a two-tailed t test yields p values of 0.0068, 0.00002 and 0.00004, respectively (both of these t tests only take into account RMS error and not chain coverage).

Figure 10b shows that the percent of the chain covered was roughly equivalent for the three approaches. However, Figure 10c shows that our approach is much better than the others at identifying the proper residue type at a particular location. However, it is important to point out that these related methods are not optimizing residue-identification accuracy. Resolve, for example, will often return a long chain of alanine residues if it cannot identify sidechains, but still gives the correct backbone structure overall. This illustrates a significant difference between ACMI and these alternate approaches: TEXTAL and Resolve build a backbone model, then attempt to align the protein sequence to it. ACMI, alternatively, *uses the sequence of the protein to construct the model*. The result is better identification of amino acids in the map.

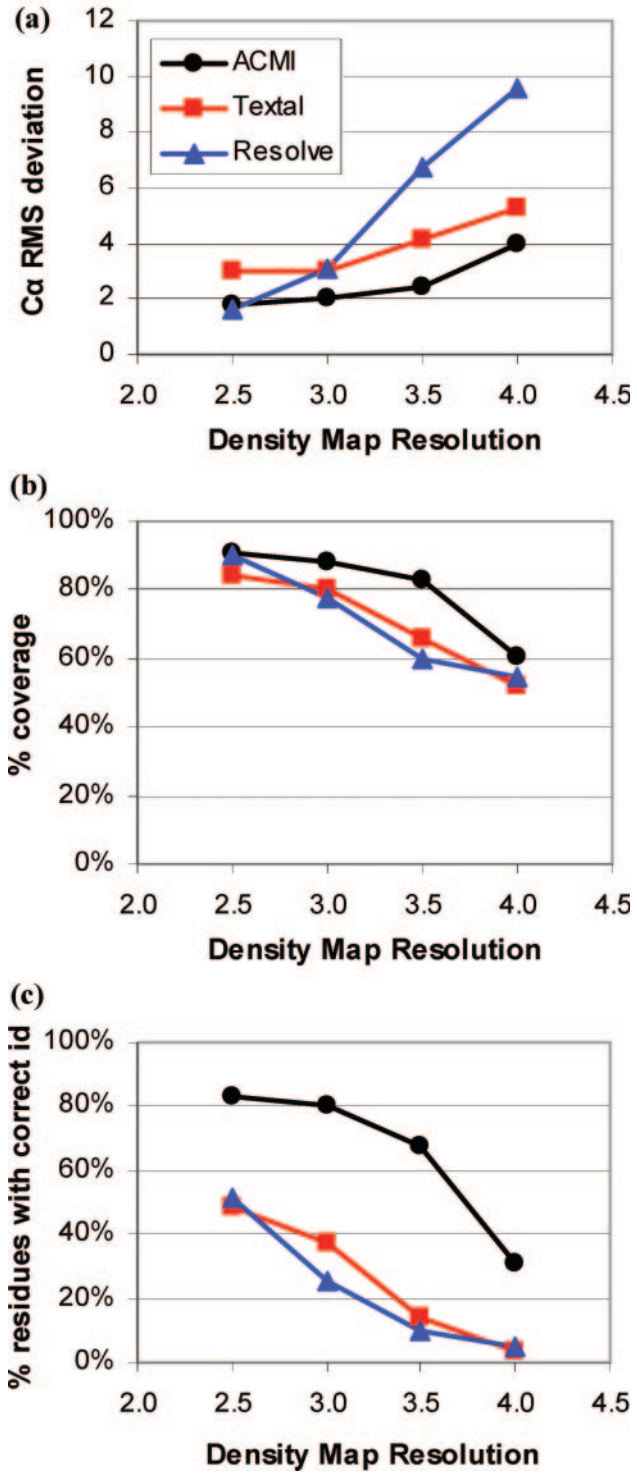


Fig. 10. Graphs showing a comparison of the three algorithms' average interpretation in terms of (a) RMS Error, (b) percent of the chain located, and (c) percent of residues correctly identified.

Additionally, Figure 11 shows scatterplots in which each individually solved electron density map is a point. The *x*-axis indicates ACMI's error; the *y*-axis indicates TEXTAL's (or Resolve's). All points above the diagonal line correspond to maps where ACMI outperformed TEXTAL (or Resolve). On the majority of structures, our interpretation has a lower RMS error then both of the other algorithms. ACMI is outperformed by Resolve on some high-resolution maps, however, ACMI currently does not perform any post-processing on predicted backbones (e.g. real-space refinement, energy minimization); also, residues are restricted on a grid, limiting accuracy to the grid spacing.

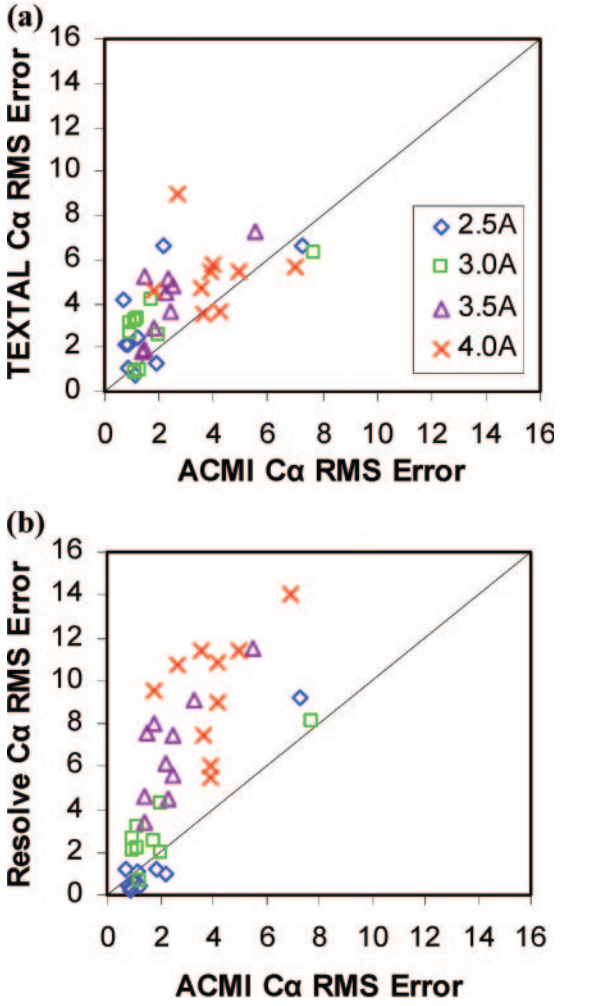


Fig. 11. A scatterplot showing the performance on a protein-by-protein basis, of ACMI versus (a) TEXTAL and (b) Resolve. Each mark is an interpreted map; points above the diagonal are cases where ACMI provided a more-accurate backbone trace.

One advantage of ACMI's probabilistic framework is that, in addition to returning a putative trace, ACMI also returns a confidence (i.e. probability) level of each predicted residue. This confidence informs the crystallographer what areas in the map need improvement; alternatively, a high confidence partial trace could be used to improve phasing. Figure 12 illustrates this in an example trace at 3.5 Å resolution, on an structure consisting of two chains of 124 residues each. This is our sixth-best (of ten) traces at this resolution: ACMI finds nine segments with a C_α RMS deviation of 2.3 Å, covering 94% of the backbone. The trace's color indicates the likelihood of its prediction for each residue's location.

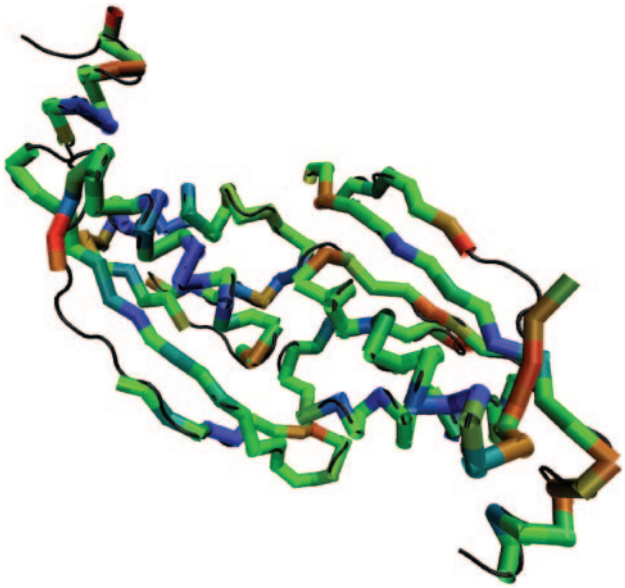


Fig. 12. A comparison of predicted versus actual structure on our sixth-best (of ten) interpretation at 3.5Å resolution. The thin continuous coil is the actual structure, while the thicker segmented chain is ACMI's prediction. The predicted structure is colored by log-likelihood, where least-likely residues are shown in red, and most-likely in blue.

7 CONCLUSIONS AND FUTURE WORK

We describe ACMI, a tool for automatically tracing protein backbones especially designed for poor-quality electron density maps. ACMI combines a local matching procedure and a global constraint procedure in a probabilistic framework that can efficiently infer the locations of backbone atoms in an electron density map. The algorithm provides accurate traces even in poor resolution electron density maps, outperforming both TEXTAL and Resolve above 3 Å map resolution.

One major shortcoming of ACMI is the significant compute time required by its local (5-mer) matching procedure. We need to search for approximately three 5-mer fragments per residue; for each fragment we consider $\sim$1900 rotations. Even for medium-sized unit cells, this takes on the order of CPU-weeks; larger proteins take months. ACMI exploits parallelism, running overnight, using the spare cycles from desktop computers [15]. However, we would like to investigate the use of machine learning algorithms, such as support vector machines or neural networks, to *quickly* match a 5-mer into the density map. We also would like to explore alternative feature representations.

Additionally, as a post-processing step, we would like to augment ACMI with a refinement and sidechain tracing algorithm. In our previous work, we used pictorial structures to place sidechain atoms, given a C_α trace [23]: combining this tool with ACMI would produce a complete molecular model.

Finally, we would like to explore the use of our probabilistic model for phase improvement. In some maps, initial phasing is quite poor. In these maps, a partial structure can be used to significantly improve the initial phasing, revealing previously blurred-out regions in the electron density. Using a high-confidence trace to iteratively improve phasing is a future research direction of ACMI.

By providing accurate interpretations from lower-resolution maps, ACMI reduces the burden on crystallographers when only poor-quality density map data is available. Even when obtaining higher-resolution electron density map data is possible, ACMI allows significant cost savings by making do with poorer-quality maps, speeding up the process of high-throughput protein structure determination.

ACKNOWLEDGEMENTS

We acknowledge support from the National Library of Medicine training grant T15-LM007359 (FD), NLM R01-LM008796 (JS), and National Institutes of Health Protein Structure Initiative Grant GM074901 (GP).

REFERENCES

[1] Berman,H. and Westbrook,J. The impact of structural genomics on the protein data bank. *Am. J. Pharmacogenomics*. 4(4).
[2] Perrakis,A., Sixma,T., Wilson,K. and Lamzin,V. (1997) wARP: Improvement and extension of crystallographic phases. *Acta Crystallographica*, D53.
[3] Terwilliger,T. (2002) Automated main-chain model-building by template-matching and iterative fragment extension. *Acta Crystallographica*, D59.
[4] Ioerger,T. and Sacchettini,J. (2003) The TEXTAL system: Artificial intelligence techniques for automated protein model building. *Methods in Enzymology*, 374.
[5] MacKerell,A.,Jr., Wiórkiewicz-Kuczera,J. and Karplus,M. (1995) An all-atom empirical energy function for the simulation of nucleic acids. *J. Am. Chem. Soc.*, 117.
[6] Terwilliger,T. (2002) Automated side-chain model-building and sequence assignment by template-matching. *Acta Crystallographica*, D59.
[7] Greer,J. (1974) Three-dimensional pattern recognition. *J. Molecular Biology*, 82.
[8] Leherte,L., Glasgow,J., Baxter,K., Steeg,E. and Fortier,S. (1997) Analysis of three-dimensional protein images. *Journal of AI Research*, 7.
[9] Cowtan,K. (2001) Fast Fourier feature recognition. *Acta Crystallographica*, D57.
[10] Wang,G. and Dunbrack,R. (2003) PISCES: A protein sequence culling server. *Bioinformatics*, 19.
[11] Jones,D., Taylor,W. and Thornton,J. (1992) The rapid generation of mutation data matrices from protein sequences. *CABIOS*, 8.
[12] Kabsch,W. (1976) A solution for the best rotation to relate two sets of vectors. *Acta. Crystallographica*, A32.
[13] Huang,S. and Hwang,J. (2004) Computation of conformational entropy from protein sequences. *Proteins: Structure, Function, and Bioinformatics*, 59(4).
[14] Cowtan,K. (1998) Modified phased translation functions and their application to molecular-fragment location. *Acta Crystallographica*, D54.
[15] Thain,D., Tannenbaum,T. and Livny,M. (2005) Distributed Computing in Practice: The Condor Experience. *Concurrency and Computation: Practice and Experience*, 17(2–4).
[16] Geman,S. and Geman,D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *PAMI*, 6.
[17] Sudderth,E., Mandel,M., Freeman,W. and Willsky,A. (2004) Visual hand tracking using nonparametric belief propagation. *MIT LIDS Technical Report 2603*.
[18] Pearl,J. (1988) *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, San Mateo.
[19] Weiss,Y. and Freeman,W.T. (2001) Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Comp.*, 13.
[20] Murphy,K., Weiss,Y. and Jordan,M. (1999) Loopy belief propagation for approximate inference. *Proc. UAI*.
[21] DiMaio,F. and Shavlik,J. (2006) Improving the Effciency of Belief Propagation in Large, Highly-Connected Graphs. *Working Paper 06-1, UW ML Research Group*.
[22] Ihler,A., Fisher,J. and Willsky,A. (2004) Message Errors in Belief Propagation. *Proc. NIPS*.
[23] DiMaio,F., Shavlik,J. and Phillips,G. (2004) Pictorial structures for molecular modeling: Interpreting density maps. *Proc. NIPS*.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e90–e98
doi:10.1093/bioinformatics/btl246

CONTRAfold: RNA secondary structure prediction without physics-based models

Chuong B. Do[1,*], Daniel A. Woods[1] and Serafim Batzoglou[1]

[1]Computer Science Department, Stanford University, Stanford, CA 94305, USA

ABSTRACT

Motivation: For several decades, free energy minimization methods have been the dominant strategy for single sequence RNA secondary structure prediction. More recently, stochastic context-free grammars (SCFGs) have emerged as an alternative probabilistic methodology for modeling RNA structure. Unlike physics-based methods, which rely on thousands of experimentally-measured thermodynamic parameters, SCFGs use fully-automated statistical learning algorithms to derive model parameters. Despite this advantage, however, probabilistic methods have not replaced free energy minimization methods as the tool of choice for secondary structure prediction, as the accuracies of the best current SCFGs have yet to match those of the best physics-based models.

Results: In this paper, we present CONTRAfold, a novel secondary structure prediction method based on *conditional log-linear models* (CLLMs), a flexible class of probabilistic models which generalize upon SCFGs by using discriminative training and feature-rich scoring. In a series of cross-validation experiments, we show that grammar-based secondary structure prediction methods formulated as CLLMs consistently outperform their SCFG analogs. Furthermore, CONTRAfold, a CLLM incorporating most of the features found in typical thermodynamic models, achieves the highest single sequence prediction accuracies to date, outperforming currently available probabilistic and physics-based techniques. Our result thus closes the gap between probabilistic and thermodynamic models, demonstrating that statistical learning procedures provide an effective alternative to empirical measurement of thermodynamic parameters for RNA secondary structure prediction.

Availability: Source code for CONTRAfold is available at http://contra.stanford.edu/contrafold/.

Contact: chuongdo@cs.stanford.edu

1 INTRODUCTION

In many RNA-related studies—ranging from noncoding RNA detection [13] to folding dynamics simulations [24] to hybridization stability assessment for microarray oligo probe selection [19]—knowing the secondary structure of an RNA sequence reveals important constraints governing the molecule's physical properties and function. To date, experimental assays for base-pairing in RNA sequences constitute the most reliable method for secondary structure determination [3]; however, their difficulty and expense are often prohibitive, especially for high-throughput applications. For this reason, computational prediction provides an attractive alternative to empirical discovery of RNA secondary structure [4].

Traditionally, the most successful techniques for single sequence computational secondary structure prediction have relied on physics models of RNA structure. Methods belonging to this category identify candidate structures for an RNA sequence by free energy minimization [22] through dynamic programming (e.g., Mfold [26] and ViennaRNA [7]) or alternative optimization schemes (e.g., RDfolder [25]).

Parameters used in energy-based methods typically come from empirical studies of RNA structural energetics. For example, parameters for nearest neighbor interactions in stacking base pairs are derived from melting curves of synthesized oligonucleotides [23]. In some cases, however, the difficulty of experimental procedures places severe restrictions on what parameters are measurable, and hence, the scoring models used. For instance, most secondary structure programs ignore the sequence dependence of hairpin, bulge, internal, and multi-branch loop energies due to the inability to quantify these effects experimentally. Similarly, the energies of multi-branch loops in modern secondary structure prediction programs rely on ad hoc scoring rules due to the lack of experimental techniques for assessing their free energy contribution [11].

Recently, stochastic context-free grammars (SCFGs) have emerged as an alternative probabilistic methodology for modeling RNA structure [2,8,9]. These models specify formal grammar rules that induce a joint probability distribution over possible RNA structures and sequences. In particular, the parameters of SCFG models specify probability distributions over possible transformations that may be applied to a ''nonterminal'' symbol, and thus are subject to the standard mathematical constraints of probability distributions (i.e. parameters may not be negative, and certain sets of parameters must sum to one). Though these parameters do not have direct physical interpretations, they are easily learned from collections of RNA sequences annotated with known secondary structures, without the need for external laboratory experiments [1].

While fairly simple SCFGs achieve respectable prediction accuracies, attempts in recent years to improve their performance using more sophisticated models have thus far yielded only modest gains. As a result, a significant performance separation still remains between the best physics-based methods and the best SCFGs [1]. Consequently, one might assume that such a gap is the inevitable price to be paid for using easily learnable probabilistic models, which are unable to provide an adequate representation of the physics underlying RNA structural stability. We assert that this is not the case.

In this paper, we present CONTRAfold, a new secondary structure prediction tool based on a flexible probabilistic model called a *conditional log-linear model* (CLLM). CLLMs generalize upon SCFGs in the sense that any SCFG has an equivalent representation

*To whom correspondence should be addressed.

as an appropriately parameterized CLLM. Like SCFGs, CLLMs enjoy the ease of computationally-driven parameter learning. Unlike vanilla SCFGs, however, CLLMs also have the generality to represent complex scoring schemes, such as those used in modern energy-based secondary structure predictors such as Mfold. CONTRAfold, a CLLM based on a simplified Mfold-like scoring scheme, not only achieves the highest single sequence prediction accuracies to date but also provides users with a new mechanism for controlling the sensitivity and specificity of the prediction algorithm.

2 METHODS

In this section, we motivate the use of CLLMs for RNA secondary structure prediction by showing how they arise as a natural extension of SCFGs. We then describe the CONTRAfold secondary structure model, which extends and simplifies traditional energy-based scoring schemes while retaining the parameter learning ease of common probabilistic methods. Finally, we describe a maximum expected accuracy decoding algorithm for secondary structure prediction which allows the user to adjust the desired sensitivity/specificity of the returned predictions via a single parameter γ.

2.1 Modeling secondary structure with SCFGs

In the RNA secondary structure prediction problem, we are given an input sequence x, and our goal is to predict the best structure y. For probabilistic parsing techniques, this requires a way to calculate the conditional probability $P(y \mid x)$ of the structure y given the sequence x.

2.1.1 Representation Stochastic context-free grammars (SCFGs) provide a compact representation of a joint probability distribution over RNA sequences and their secondary structures. An SCFG for secondary structure prediction defines (1) a set of transformation rules, (2) a probability distribution over the transformation rules applicable to each nonterminal symbol, and (3) a mapping from parses (derivations) to secondary structures.

For example, consider the following simple unambiguous SCFG for a restricted class of RNA secondary structures:

(1) *Transformation rules*.

$$S \rightarrow \mathtt{a}S\mathtt{u} \mid \mathtt{u}S\mathtt{a} \mid \mathtt{c}S\mathtt{g} \mid \mathtt{g}S\mathtt{c} \mid \mathtt{g}S\mathtt{u} \mid \mathtt{u}S\mathtt{g} \mid \mathtt{a}S \mid \mathtt{c}S \mid \mathtt{g}S \mid \mathtt{u}S \mid \epsilon.$$

(2) *Rule probabilities*. The probability of transforming a nonterminal S into $\mathtt{a}S\mathtt{u}$ is $p_{S\rightarrow \mathtt{a}S\mathtt{u}}$, and similarly for the other transformation rules.

(3) *Mapping from parses to structures*. The secondary structure y corresponding to a parse σ contains a base pairing between two letters if and only if the two letters were generated in the same step of the derivation for σ.

For a sequence $x = \mathtt{agucu}$ with secondary structure[1] $y = (\,(\,.\,)\,)$, the unique parse σ corresponding to y is

$$S \rightarrow \mathtt{a}S\mathtt{u} \rightarrow \mathtt{ag}S\mathtt{cu} \rightarrow \mathtt{agu}S\mathtt{cu} \rightarrow \mathtt{agucu}. \quad (1)$$

The SCFG models the *joint* probability of generating the parse σ and the sequence x as

$$P(x,\sigma) = p_{S\rightarrow \mathtt{a}S\mathtt{u}} \cdot p_{S\rightarrow \mathtt{g}S\mathtt{c}} \cdot p_{S\rightarrow \mathtt{u}S} \cdot p_{S\rightarrow \epsilon}. \quad (2)$$

It follows that[2]

$$P(y \mid x) = \sum_{\sigma \in y} P(\sigma \mid x) = \frac{\sum_{\sigma \in y} P(x,\sigma)}{\sum_{\sigma' \in \Omega(x)} P(x,\sigma')}, \quad (3)$$

where $\Omega(x)$ is the space of all possible parses of x.

[1]The secondary structure of a sequence can be represented in *nested parenthesis* format, in which pairs of matching parentheses represent base pairings in the sequence.

[2]Here, we regard y as a ''set'' of parses σ sharing the same secondary structure. Note that in ambiguous grammars, the mapping from parses to secondary structures may be many-to-one.

2.1.2 Parameter estimation One of the chief advantages of SCFGs as a language for describing RNA secondary structure is the existence of well-understood algorithms for parameter estimation. Given a set $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ of m pairs of RNA sequences $x^{(i)}$ with experimentally-validated secondary structures $y^{(i)}$, the training task involves finding the set of parameters $\boldsymbol{\theta} = \{p_1, \ldots, p_n\}$ (i.e., the probabilities for each of the n transformation rules) that maximize some specified objective function.

In the popular maximum likelihood approach, $\boldsymbol{\theta}$ is chosen to maximize the *joint likelihood* of the training sequences and their structures,

$$\ell_{\mathrm{ML}}(\boldsymbol{\theta} : \mathcal{D}) = \prod_{i=1}^{m} P(x^{(i)}, y^{(i)}; \boldsymbol{\theta}), \quad (4)$$

subject to the contraints that all parameters must be nonnegative, and certain group of parameters must sum to one. For unambiguous grammars, the solution $\boldsymbol{\theta}_{\mathrm{ML}}$ to this constrained optimization problem exists in closed form. Consequently, the maximum likelihood technique is by far the most commonly used method for SCFG parameter estimation in practice.

2.2 From SCFGs to CLLMs

Like SCFGs, *conditional log-linear models* (CLLMs) are probabilistic models which have the goal of defining the conditional probability of an RNA secondary structure y given a sequence x. Here, we motivate the CLLM framework by comparison to SCFGs.

2.2.1 Representation To understand how CLLMs generalize upon the representation of conditional probabilities for SCFGs, we first consider a feature-based representation of SCFGs that highlights several important assumptions made when modeling with SCFGs. Removing these assumptions leads directly to the CLLM framework.

For a particular parse σ of a sequence x, let $\mathbf{F}(x,\sigma) \in \mathbb{R}^n$ be an n-dimensional *feature vector* (where n is the number of rules in the grammar) whose ith dimension, $F_i(x,\sigma)$, indicates the number of times the ith transformation rule is used in parse σ. Furthermore, let p_i denote the probability for the ith transformation rule. We rewrite the joint likelihood of the sequence x and its parse σ in *log-linear* form as

$$\begin{aligned} P(x,\sigma) &= \prod_{i=1}^{n} p_i^{F_i(x,\sigma)} = \exp\left(\ln\left(\prod_{i=1}^{n} p_i^{F_i(x,\sigma)}\right)\right) \\ &= \exp\left(\sum_{i=1}^{n} F_i(x,\sigma) \ln p_i\right) = \exp(\mathbf{w}^T \mathbf{F}(x,\sigma)), \end{aligned} \quad (5)$$

where $w_i = \ln p_i$. Substituting this form into equation 3,

$$P(y \mid x) = \frac{\sum_{\sigma \in y} \exp(\mathbf{w}^T \mathbf{F}(x,\sigma))}{\sum_{\sigma' \in \Omega(x)} \exp(\mathbf{w}^T \mathbf{F}(x,\sigma'))}. \quad (6)$$

In this alternate form, we see that SCFGs are actually log-linear models with the restrictions that

(1) the parameters $w_1, \ldots, w_n$ correspond to log probabilities and hence obey a number of constraints (e.g., all parameters must be negative), and

(2) the features $F_1(x,\sigma), \ldots, F_n(x,\sigma)$ derive directly from the grammar; thus the types of features are restricted by the complexity of the grammar.

In both cases, the imposed restriction is unnecessary if we simply wish to ensure that the conditional probability in equation 6 is well-defined. Removing these restrictions, thus, is the basis for the CLLM framework. More generally, CLLMs are probabilistic models defined by equation 6, in the case that the parameters $w_1, \ldots, w_n$ may take on any real values, and the feature vectors are similarly unrestricted.[3]

[3]Note that conditional random fields (CRFs) are a specialized class of CLLMs whose probability distributions are defined in terms of graphical models [10].

2.2.2 Parameter estimation By definition, CLLMs parameterize the conditional probability $P(y \mid x)$ as a log linear function of the model's features $\mathbf{F}(x, \sigma)$, but they provide no manner for calculating $P(x, y)$. As a side effect, straight maximum likelihood techniques, which optimize this joint probability, do not apply to CLLMs.

Instead, CLLM training relies on the *conditional maximum likelihood* principle, in which one finds the parameters $\mathbf{w}_{\mathrm{CML}} \in \mathbb{R}^n$ that maximize the *conditional likelihood*[4] of the structures given the sequences,

$$\ell_{\mathrm{CML}}(\mathbf{w} : \mathcal{D}) = \prod_{i=1}^{m} P(y^{(i)} \mid x^{(i)}; \mathbf{w}). \tag{7}$$

Arguably, for prediction problems, conditional likelihood (or *discriminative*) training is more natural than joint likelihood (or *generative*) training as it focuses on finding parameters that give good predictive performance without attempting to model the distribution over input sequences x.

The mechanics of performing the probabilistic inference tasks required in the optimization of equation 7 follow closely the traditional inside and outside algorithms for SCFGs [2].

2.3 From energy-based models to CLLMs

Converting an SCFG to a CLLM by removing restrictions on the parameter vector $\mathbf{w}$ and training via conditional likelihood allows SCFGs to obtain many of the benefits of the discriminative learning approach. Straightforward conversions of this sort are routine in the machine learning literature and have recently been applied to RNA secondary structure alignment [21]. Such conversions, however, do not take full advantage of the expressivity of CLLMs. In particular, the ability of CLLMs to use generic feature representations means that in some cases, CLLMs can conveniently represent models which do not have compact parameterizations as SCFGs.

For example, the QRNA algorithm [18] attempts to capture the salient properties of standard thermodynamic models for RNA secondary structure, such as loop lengths and base-stacking, via an SCFG. This conversion, however, is only approximate. In particular, the usual energy rules [23,11] contain *terminal mismatch* terms describing the interaction between closing base pairs of helices and nucleotides in the adjacent loop. These interactions are ignored in QRNA, and more generally, are difficult to incorporate in SCFG models without considerably increasing grammar complexity. As the authors themselves note, QRNA underperforms compared to standard folders, highlighting the difficulty of building SCFGs on par with energy-based methods [18].

Contrastingly, the complex scoring terms of thermodynamic models transfer to CLLMs with no difficulties. In the standard model, the energy of a folding σ decomposes as the sum of energies for hairpin, interior, bulge, stacking pair, and multi-branch loops. In turn, the energy of each type of loop further decomposes as the sum of interaction energies over individual features of the sequence x and its parse σ. Thus, in the CLLM equivalent of standard thermodynamic scoring, the parameters $w_1, \ldots, w_n$ replace the interaction energy contributions for various secondary elements, and the features $F_1(x, \sigma), \ldots, F_n(x, \sigma)$ count the number of times a particular interaction term appears in the parse σ. This procedure is illustrated in Figures 1 and 2.

2.4 The CONTRAfold model

The CONTRAfold program implements a CLLM for RNA secondary structure prediction, following the general strategy for model construction outlined in the previous section. The features in CONTRAfold (see Figure 3) include:

(1) base pairs,
(2) helix closing base pairs,

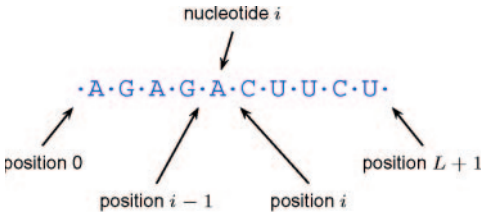


Fig. 1. Positions in a sequence of length $L = 10$. Here, let x_i denote the ith nucleotide of x. For ease of notation, we say that there are $L + 1$ *positions* corresponding to x—one position at each of the two ends of x, and $L - 1$ positions between consecutive nucleotides of x. We assign indices ranging from 0 to L for each position.

(3) hairpin lengths,
(4) helix lengths,
(5) bulge loop lengths,
(6) internal loop lengths,
(7) internal loop asymmetry,
(8) full two-dimensional table of internal loop scores,
(9) helix base pair stacking interactions,
(10) terminal mismatch interactions,
(11) single (dangling) base stacking,
(12) affine multi-branch loop scoring, and
(13) free bases.

To a large extent, the features above closely mirror the features employed in traditional thermodynamic models of RNA secondary structure. We point out a few key differences:

(1) CONTRAfold makes use of generic feature sets without incorporating ''special cases'' typical of complex thermodynamic scoring models, such as the popular Turner energy rules [11]. For instance, CONTRAfold
   – omits the bonus free energies for special case hairpin loops (specifically items (d) through (f) from the list in Figure 2).
   – does not contain a table exhaustively enumerating all possible 1×1, 1×2, 2×2, and 2×3 internal loops.

   While such features may be useful, they are more likely to lead to overfitting due to the large number of parameters that must be trained.[5] Incorporation of a small number of specially selected interactions which are known to be particularly important *a priori* is more feasible.

(2) Internal and bulge loop lengths are scored separately as a function of the lengths ℓ_1 and ℓ_2 of each side of the loop:

$$f_{\text{single length}}(\ell_1, \ell_2) = \begin{cases} w_{\text{bulge length}}[\ell_1 + \ell_2] & \text{if } \ell_1 \ell_2 = 0 \\ w_{\text{internal length}}[\ell_1 + \ell_2] & \text{otherwise} \\ \quad + w_{\text{internal asymmetry}}[|\ell_1 - \ell_2|] & \\ \quad + w_{\text{internal correction}}[\ell_1][\ell_2]. & \end{cases} \tag{8}$$

   In most thermodynamic models, only bulge and internal loop length score tables exist, whereas internal loop asymmetry is scored according to the Ninio equations [14]. Here, CONTRAfold learns an explicit scoring table $w_{\text{internal asymmetry}}[\cdot]$ for internal loop asymmetry in addition to a two-dimensional correction matrix $w_{\text{internal correction}}[\cdot][\cdot]$ for representing dependencies not captured by total loop length and asymmetry alone.

[4] In practice, we avoid overfitting by placing a zero-mean Gaussian regularization prior on the parameters, and selecting the variance of the prior using holdout cross-validation on training data only (see Results).

[5] This may be considered an advantage of physics-based methods; a hybrid approach which combines machine learning with physics-based prior knowledge may help alleviate the burden on the learning algorithm.

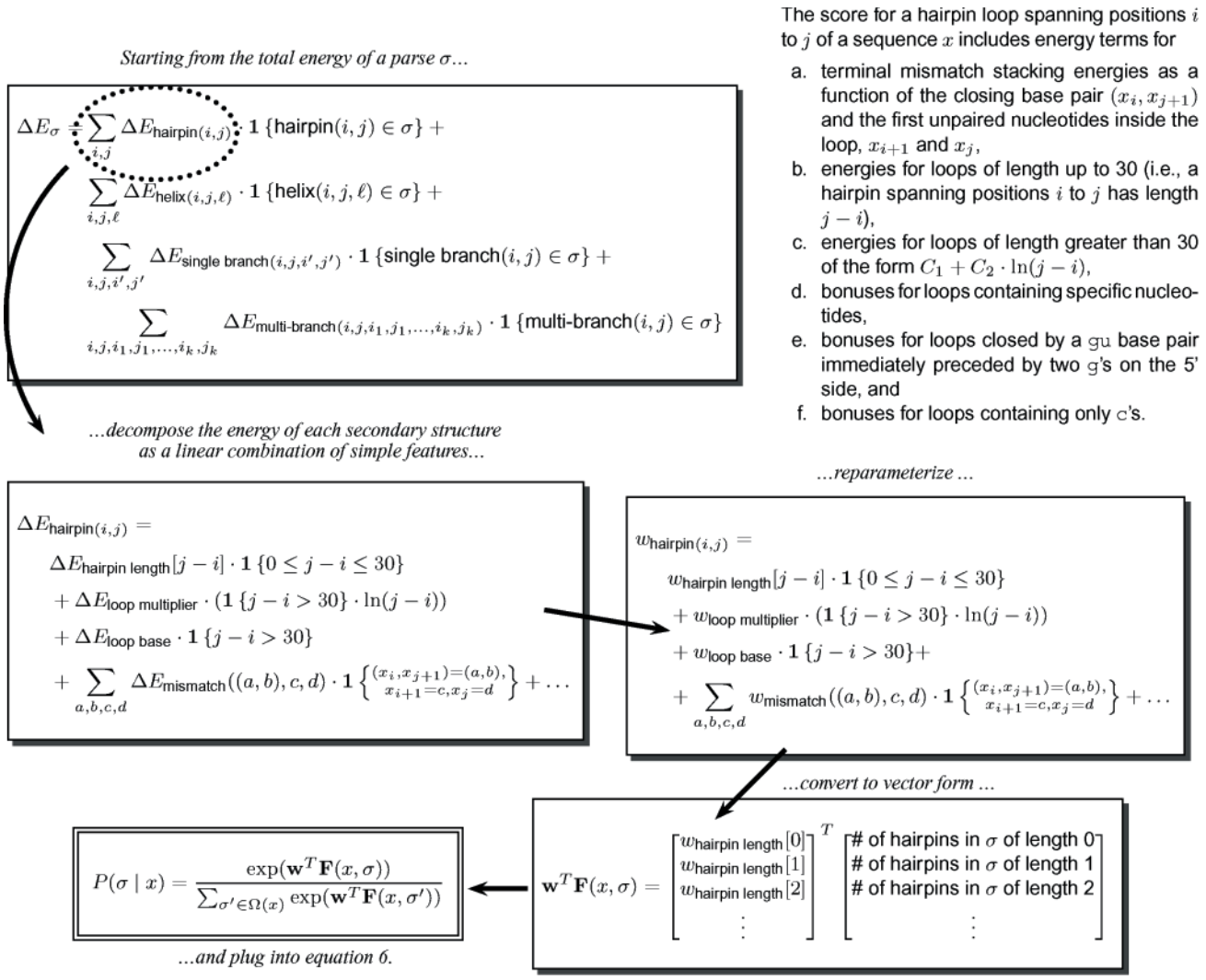


Fig. 2. The construction of a CLLM from an energy-based model. In short, the conversion process involves expressing the total energy of a parse σ as a linear function of counts for joint features $F_i(x, \sigma)$ of the sequence x and the parse σ. Once this is done, substituting into equation 6 gives a probabilistic model whose Viterbi parse is the minimum energy parse.

(3) Unlike typical energy minimization schemes, the energy of a helix consists not only of stacking interactions but also direct base pair interactions. Also, all combinations of nucleotide pairs are allowed, unlike the standard nearest neighbor model in which only canonical Watson-Crick or wobble gu pairs are permitted. Finally, CONTRAfold introduces new scoring terms for helix lengths (via an explicit scoring table for helices of length up to 5 and affine afterwards), which are not part of the standard nearest neighbor model.

(4) Since little is currently known about the energetics of free bases (bases which do not belong to any other loop in the secondary structure), they are typically ignored by energy-based folders. Here, CONTRAfold introduces two scoring parameters: $w_{\text{outer unpaired}}$ for scoring each free base, and $w_{\text{outer paired}}$ for scoring each base pair adjacent to a free base.

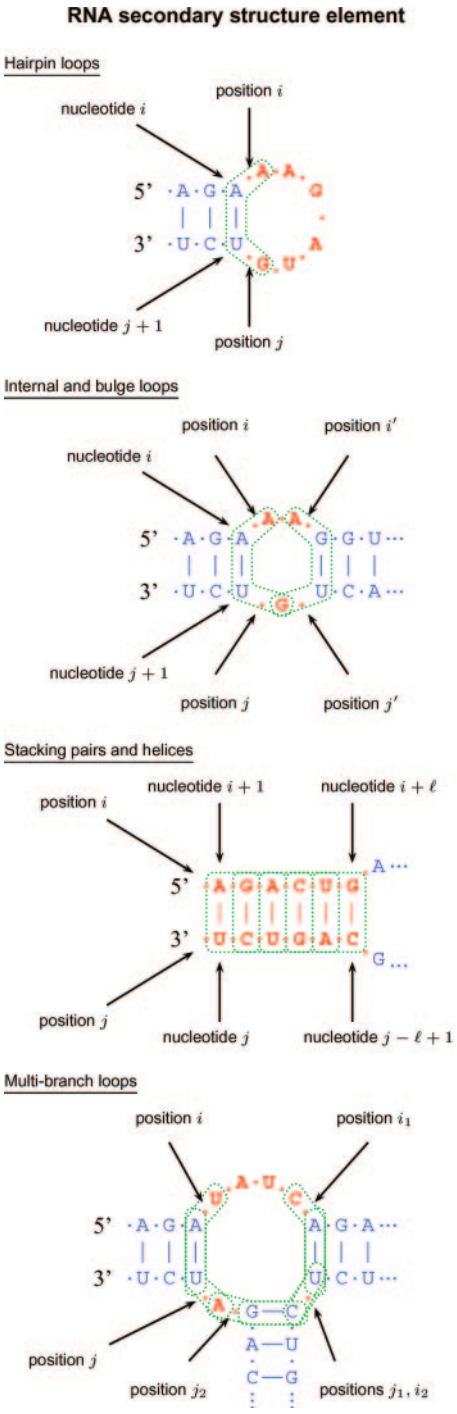
(5) For simplicity, CONTRAfold scores terminal mismatches for hairpins, bulges, and internal loops using the same parameters. CONTRAfold also does not account for coaxial stacking dependencies when scoring multi-branch loops. Like the special case hairpin loops mentioned earlier, making more specific scoring models by differentiating between these terminal mismatches may improve prediction accuracy.

2.5 Maximum expected accuracy parsing with sensitivity/specificity tradeoff

Most physics-based approaches to secondary structure prediction use dynamic programming to recover the structure with minimum free energy [26,7]. For probabilistic methods, the Viterbi algorithm (known as the CYK algorithm [2] for SCFGs) fulfills this function by finding the most likely parse,[6]

$$\hat{\boldsymbol{\sigma}}_{\text{viterbi}} = \arg\max_{\boldsymbol{\sigma} \in \Omega(x)} P(\boldsymbol{\sigma} \mid x; \mathbf{w}). \quad (9)$$

[6]For unambiguous grammars, the most likely parse is also the most likely secondary structure; however, this is not the case for ambiguous grammars [1,16].



Main components of CLLM scoring

$$w_{\text{hairpin}}(i,j) = w_{\text{terminal mismatch}}((x_i, x_{j+1}), x_{i+1}, x_j) + \begin{cases} w_{\text{hairpin length}}[j-i] & \text{if } 0 \le j-i \le 30 \\ w_{\text{loop base}} + w_{\text{loop multiplier}} \cdot \ln(j-i) & \text{if } j-i > 30. \end{cases}$$

Hairpin loop scoring in CONTRAfold accounts for terminal mismatch interactions between the hairpin closing base pair and the adjacent nucleotides in the hairpin loop (see left diagram), and hairpin lengths. Following energy-based models, we set the score for hairpins of length beyond 30 to be affine in the logarithm of the hairpin size.

$$\begin{aligned} w_{\text{single}}(i,j,i',j') &= f_{\text{single length}}(i'-i, j-j') \\ &+ w_{\text{terminal mismatch}}((x_i, x_{j+1}), x_{i+1}, x_j) \\ &+ w_{\text{terminal mismatch}}((x_{j'}, x_{i'+1}), x_{j'+1}, x_{i'}) \end{aligned}$$

Internal and bulge loops (also known as single branch loops) have exactly two adjacent base pairs. CONTRAfold scores internal and bulge loops by accounting for loop lengths and terminal mismatches (see left diagram). Loop lengths are scored according to equation 8. Terminal mismatches are scored in the same way as terminal mismatches in hairpin loops.

$$\begin{aligned} w_{\text{helix}}(i,j,\ell) &= \textstyle\sum_{k=1}^{\ell} w_{\text{base pair}}((x_{i+k}, x_{j+1-k})) \\ &+ \textstyle\sum_{k=1}^{\ell-1} w_{\text{stacking}}((x_{i+k}, x_{j+1-k}), (x_{i+k+1}, x_{j-k})) \\ &+ w_{\text{helix closing}}(x_{i+1}, x_j) + w_{\text{helix closing}}(x_{j-\ell+1}, x_{i+\ell}) \end{aligned}$$

CONTRAfold scores helices (regions of adjacent stacking base pairs) by including terms for the contribution of closing base pairs as well as each stacking and base pair interaction. Unlike the standard thermodynamic model, which allows only canonical Watson-Crick or wobble gu pairs, CONTRAfold allows for all possible nucleotide combinations.

$$\begin{aligned} w_{\text{multi}}(i,j,i_1,j_1,\ldots,i_m,j_m) &= w_{\text{multi base}} \\ &+ w_{\text{multi paired}} \cdot (m+1) + w_{\text{multi unpaired}} \cdot \ell \\ &+ w_{\text{stacking left}}((x_i, x_{j+1}), x_{i+1}) + w_{\text{stacking right}}((x_i, x_{j+1}), x_j) \\ &+ \textstyle\sum_{k=1}^{m} w_{\text{stacking left}}((x_{j_k}, x_{i_k+1}), x_{j_k+1}) \\ &+ \textstyle\sum_{k=1}^{m} w_{\text{stacking right}}((x_{j_k}, x_{i_k+1}), x_{i_k}) \end{aligned}$$

Multi-branch loops have at least three adjacent base pairs. Scoring of multi-branch loops in CONTRAfold accounts for the single base stacking interactions and the length dependence of multi-branch loops. For computational efficiency reasons [11], multi-branch loop length scores are affine in the number of unpaired nucleotides (ℓ) and the number of adjacent base pairs ($m+1$).

Fig. 3. Correspondence between energy-based model scoring and CLLM potentials in CONTRAfold. In each diagram, the nucleotides comprising the indicated RNA secondary structure element are shown in red. Green dotted lines indicate the groups of nucleotides involved in the terminal mismatch, helix stacking, or single base stacking interactions considered by CONTRAfold.

Here, we describe an alternative scheme that, for a given setting of a sensitivity/specificity tradeoff parameter γ, identifies the structure with *maximum expected accuracy*.

In particular, for a candidate structure $\hat{y}$ with true structure y, let $\text{accuracy}_\gamma(\hat{y}, y)$ denote the number of correctly unpaired positions in $\hat{y}$ (with respect to y) plus γ times the number of correctly paired positions in $\hat{y}$. Then, we wish to find,

$$\hat{y}_{\text{mea}} = \arg\max_{\hat{y}} \mathbb{E}_y[\text{accuracy}_\gamma(\hat{y}, y)], \tag{10}$$

where the expectation is taken with respect to the conditional distribution over structures of the sequence x.

To do this, let p_{ij} denote the conditional probability that the ith and jth nucleotides of sequence x base pair. Similarly, let $q_i = 1 - \sum_j p_{ij}$ be the conditional probability that the ith nucleotide is unpaired. The following recurrence computes $M_{1,L} = \max_y(E_y[\text{accuracy}_\gamma(\hat{y}_{\text{mea}}, y)])$:

$$M_{i,j} = \max \begin{cases} q_i & \text{if } i = j \\ q_i + M_{i+1,j} & \text{if } i < j \\ q_j + M_{i,j-1} & \text{if } i < j \\ \gamma \cdot 2p_{ij} + M_{i+1,j-1} & \text{if } i + 2 \leq j \\ M_{i,k} + M_{k+1,j} & \text{if } i \leq k < j. \end{cases} \quad (11)$$

Including the traceback for recovering the optimal structure, the parsing algorithm takes $O(L^3)$ time and $O(L^2)$ space.

Note that in the above algorithm, γ controls the balance between the sensitivity and specificity of the returned structure—i.e., higher values of γ encourage the parser to predict more base pairings whereas lower values of γ restrict the parser to predicting only base pairs for which the algorithm is extremely confident. When $\gamma = 1$, the algorithm maximizes the expected number of correct positions and is identical to the parsing technique used in Pfold [9]. As shown in the Results section, by allowing γ to vary, we may adjust the sensitivity and specificity of the parsing algorithm as desired.

3 RESULTS

To assess the suitability of CLLMs as models for RNA secondary structure, we performed a series of cross-validation experiments using known consensus secondary structures of noncoding RNA families taken from the Rfam database [5,6]. Specifically, version 7.0 of Rfam contains seed multiple alignments for 503 noncoding RNA families, and consensus secondary structures for each alignment either taken from a previously published study in the literature or predicted using automated covariance-based methods.

To establish ''gold-standard'' data for training and testing, we first removed all seed alignments with only predicted secondary structures, retaining the 151 families with secondary structures from the literature. For each of these families, we then projected the consensus family structure to every sequence in the alignment, and retained the sequence/structure pair with the lowest combined proportion of missing nucleotides and non-{au, cg, gu} base pairs. The end result was a set of 151 independent examples, each taken from a different RNA family.

3.1 Comparison to generative training

In our first experiment, we took nine different grammar-based models (G1-G8, G6s) from a recent study by Dowell and Eddy on the performance of simple SCFGs for RNA secondary structure prediction [1]. For each grammar, we took the original SCFG and constructed an equivalent CLLM. We then applied a two-fold cross-validation procedure to compare the performance of SCFG (generative) and CLLM (discriminative) parameter learning.

In particular, we partitioned the 151 selected sequence-structure pairs randomly into two approximately equal-sized ''folds.'' For any given setting of the MEA trade-off parameter γ, we used parameters trained on sequences from one fold[7] to perform predictions for all sequences from the other fold. For each tested example, we computed sensitivity and specificity (PPV)[8], defined as

$$\text{sensitivity} = \frac{\text{number of correct base pairings}}{\text{number of true base pairings}} \quad (12)$$

$$\text{specificity} = \frac{\text{number of correct base pairings}}{\text{number of predicted base pairings}}. \quad (13)$$

By repeating this cross-validation procedure for values of $\gamma \in \{2^k : -5 \leq k \leq 10\}$, we obtained a receiver operating characteristic (ROC) curve for each grammar. We report the estimated area under each curve (see Table 1). In 7 out of 9 grammars, the CLLM outperforms its SCFG counterpart.

Table 1. Comparison of generative and discriminative model structure prediction accuracy.

Grammar	Generative	Discriminative	Difference
G1	0.0392	**0.2713**	+0.2321
G2	0.3640	**0.5797**	+0.2157
G3	**0.4190**	0.4159	−0.0031
G4	**0.1361**	0.1350	−0.0011
G5	0.0026	**0.0031**	+0.0005
G6	0.5446	**0.5600**	+0.0154
G7	0.5456	**0.5582**	+0.0126
G8	0.5464	**0.5515**	+0.0051
G6s	0.5501	**0.5642**	+0.0141

Each number in the table represents the area under the ROC curve of an MEA-based parser using the indicated model. As seen below, the discriminative model consistently outperforms its generative counterpart.

Using a similar cross-validation protocol, we also found that MEA parsing outperforms the Viterbi algorithm on average for both the generative and discriminative models. In particular, when an algorithm A achieves better sensitivity and specificity than algorithm B, we say that A *dominates* B. On 7 out of 9 generatively-trained grammars and 9 out of 9 discriminatively-trained grammars, we found a γ for which the MEA parsing algorithm dominates the Viterbi algorithm (see Table 2).

3.2 Comparison to other methods

Next, we compared the performance of CONTRAfold with a number of leading probabilistic and free energy minimization methods. In particular, we benchmarked Mfold v3.2 [26], ViennaRNA v1.6 [7], PKNOTS v1.05 [17][9], Pfold v3.2 [9], and ILM [20], using default parameters for each program.[10] Whenever a program returned multiple possible structures (e.g., Mfold), we scored only the structure with minimum predicted free energy.

[7]To determine smoothing parameters (for SCFGs) or regularization constants (for CLLMs), we used conditional log-likelihood on a holdout set taken from the training data as an estimate of the generalization ability of the learned model, and found the optimal setting of the desired parameter using a golden section search [15].

[8]We considered only au, cg, and gu base pairs since many of the energy-based folders cannot predict other types of base pairings as a consequence of the nearest neighbor model.

[9]Because of the large size of some of the sequences in our dataset, we disabled pseudoknot prediction for PKNOTS.

[10]Note that while all tools listed support single sequence RNA secondary structure prediction, not all were designed specifically for single sequence prediction. Pfold, for instance, was developed in the context of multiple sequence structure prediction; similarly, ILM and PKNOTS were developed for prediction of RNA structures with pseudoknots, and so might fare better on sequences where pseudoknot interactions play a more important role.

Table 2. Comparison of generative and discriminative model structure prediction accuracy

Grammar	Generative Viterbi Sens (spec)	Generative MEA Sens (spec)	Discriminative Viterbi Sens (spec)	Discriminative MEA Sens (spec)
G1	**0.41 (0.27)**	0.18 (0.11)	0.40 (0.28)	**0.48 (0.33)**
G2	0.53 (0.36)	0.53 (0.36)	0.63 (0.48)	**0.67 (0.64)**
G3	0.46 (0.48)	**0.56 (0.51)**	0.45 (0.46)	**0.54 (0.53)**
G4	0.21 (0.17)	**0.33 (0.23)**	0.21 (0.17)	**0.34 (0.23)**
G5	0.03 (0.04)	**0.06 (0.04)**	0.02 (0.03)	**0.06 (0.04)**
G6	0.60 (0.61)	**0.62 (0.63)**	0.61 (0.62)	**0.62 (0.67)**
G6s	0.60 (0.62)	**0.62 (0.64)**	0.62 (0.63)	**0.65 (0.65)**
G7	0.58 (0.63)	**0.63 (0.63)**	0.58 (0.62)	**0.63 (0.67)**
G8	0.58 (0.60)	**0.63 (0.62)**	0.58 (0.61)	**0.65 (0.62)**

In each case, γ was adjusted for MEA parsing to allow a direct comparison with Viterbi, and the dominant parsing method is shown in bold. Finally, note that the results for MEA reflect only a single choice of γ rather than the entire ROC curve, so one should refer to Table 1 for a more reliable comparison of generative and discriminative MEA accuracy.

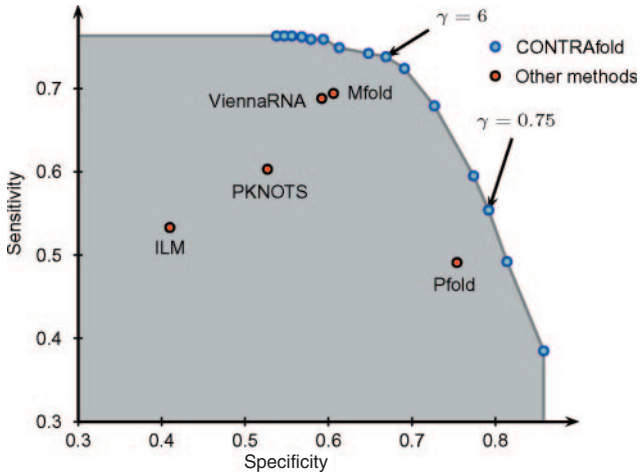


Fig. 4. ROC plot comparing sensitivity and specificity for several RNA structure prediction methods. CONTRAfold performance was measured at several different settings of the γ parameter, which controls the tradeoff between the sensitivity and specificity of the prediction algorithm. As shown above, CONTRAfold achieves the highest sensitivity at each level of specificity.

Unlike the other programs in our comparison, CONTRAfold's use of the maximum expected accuracy algorithm for parsing allows it to optimize for either higher sensitivity or higher specificity via the constant γ. In Figure 4, we varied the choice of γ for the parsing algorithm so as to allow CONTRAfold to achieve many different trade-offs between sensitivity and specificity; some of these trade-offs allow for unambiguous comparisons between CONTRAfold and existing methods.

As shown in Tables 3 and 4, CONTRAfold outperforms existing probabilistic and energy-based structure prediction methods without relying on the thousands of experimentally measured parameters common among free energy minimization techniques. For $\gamma = 6$ in particular, CONTRAfold achieves statistically significant improvements of over 4% in sensitivity and 6% in specificity relative to the best current method, Mfold. This demonstrates not only the quality of the underlying model but also the effectiveness of the parsing mechanism for providing a sensitivity/specificity trade-off.

Table 3. Accuracies of leading secondary structure prediction methods

Method	Sensitivity	Specificity	Time (s)
CONTRAfold (γ=6)	**0.7377**	**0.6686**	224
Mfold	0.6943	0.6063	62
ViennaRNA	0.6877	0.5922	**8**
PKNOTS	0.6030	0.5269	460
ILM	0.5330	0.4098	22
CONTRAfold (γ =0.75)	**0.5540**	**0.7920**	224
Pfold	0.4906	0.7535	**22**

Table 4. Performance of CONTRAfold relative to leading secondary structure prediction methods

Method	Sensitivity +	Sensitivity −	Sensitivity p-value	Specificity +	Specificity −	Specificity p-value
Mfold	34	69	0.00081	51	77	0.0271
ViennaRNA	30	72	4.9×10^{-5}	44	82	0.00098
PKNOTS	17	94	5.5×10^{-13}	26	104	1.5×10^{-11}
ILM	20	101	3.6×10^{-13}	12	126	6.8×10^{-22}
Pfold	38	72	0.0017	41	64	0.0318

Mfold, ViennaRNA, PKNOTS, and ILM were compared to CONTRAfold ($\gamma = 6$). Pfold was compared to CONTRAfold ($\gamma = 0.75$). The numbers in the +/− columns indicate the number of times the method achieved higher (+) or lower (−) sensitivity/specificity than CONTRAfold. p-values were calculated using the sign test.

3.3 Feature assessment

To understand the importance of various features to the CONTRAfold model, we performed an abrasion analysis in which we removed various sets of features from the model and assessed the change in total ROC area for the MEA parser. As seen in Table 5, the performance of CONTRAfold degrades as features are removed from the model.

Interestingly, even the weakest model from Table 5, which includes only features for hairpin, bulge, internal, multi-branch loops (without accounting for internal loop asymmetry), helix closing base pairs, and helix base pairs, achieves a respectable ROC area of 0.6003. In fact, this crippled version of CONTRAfold, which does not even account for helix stacking interactions, manages to obtain sensitivity and specificity values of 0.7006 and 0.6193, respectively, accuracy statistically indistinguishable from Mfold.

3.4 Learned versus measured parameters

In many respects, the general techniques employed by CLLMs are reminiscent of many previously described algorithms. For instance,

Table 5. Abrasion analysis of CONTRAfold model

Variant	ROC area	Decrease
CONTRAfold	0.6433	n/a
(without single base stacking)	0.6416	0.0017
(without helix lengths)	0.6370	0.0063
(without terminal mismatch penalties)	0.6362	0.0071
(without full internal loop table)	0.6336	0.0097
(without helix stacking)	0.6276	0.0157
(without outer)	0.6271	0.0162
(without internal loop asymmetry)	0.6134	0.0299
(without all of the above)	0.6003	0.0430

A large decrease in ROC area suggests that the corresponding removed features play an important role in RNA secondary structure. However, the reverse is not true: small decreases in accuracy (such as seen for single base stacking) may simply mean that CONTRAfold was less effective in leveraging that feature for prediction.

(a) Learned

$5' \longrightarrow 3'$: $\mathtt{a}X$ / $\mathtt{u}Y$; $3' \longleftarrow 5'$

X \ Y	a	c	g	u
a	0.48	0.38	0.34	-1.24
c	0.27	0.33	-1.74	0.34
g	0.34	-1.63	0.27	-0.74
u	-1.26	0.32	-0.89	0.32

(b) Experimental

$5' \longrightarrow 3'$: $\mathtt{a}X$ / $\mathtt{u}Y$; $3' \longleftarrow 5'$

X \ Y	a	c	g	u
a	.	.	.	-0.90
c	.	.	-2.20	.
g	.	-2.10	.	-0.60
u	-1.10	.	-1.40	.

Fig. 5. Comparison of learned and experimentally measured stacking energies. (**a**) A portion of the helix stacking parameters learned by CONTRAfold, scaled by $-RT$ at $T = 310.15$ K $= 37°$C. (**b**) A portion of the helix stacking energies from the Turner 3.0 energy rules [11], as taken from the Mfold package [26].

the inside-outside algorithms inspired by SCFGs bear close relation to McCaskill's procedure for computing base-pairing probabilities via the partition function [12]. Indeed, one may be tempted to draw direct analogies between the parameters of energy-based models and the parameters learned by the CLLM (appropriately scaled by $-RT$, the negated product of the universal gas constant and absolute temperature).

As shown in Figure 5, in some cases one can find a good correlation between parameters learned by CONTRAfold and those measured experimentally. Differences between learned parameters and measured values, however, are not necessarily diagnostic of errors in the laboratory measurements. Roughly speaking, the parameters learned by CLLMs reflect the degree of enrichment of their corresponding features in training set secondary structures. Therefore, parameters which do not appear often in training set structures will have smaller parameter values, regardless of their actual energetic contribution to real RNA structures. Additionally, Gaussian prior regularization (see footnote to Section 2.2.2), reduces the magnitude of less confident parameters to prevent overfitting. Finally, CLLM learning compensates for dependencies between parameters so as to maximize the overall conditional likelihood of the training set; thus, the values learned for one parameter will depend greatly on the other parameters in the model.

4 DISCUSSION

In this paper, we presented CONTRAfold, a new RNA secondary structure prediction method based on conditional log-linear models (CLLMs). Like previous structure prediction methods based on probabilistic models, CONTRAfold relies on statistical learning techniques to optimize model parameters according to a training set. Unlike its predecessors, however, CONTRAfold uses a discriminative training objective and flexible feature representations in order to achieve accuracies exceeding those of the current best physics-based structure predictors.

As a modeling framework for RNA secondary structure prediction, CLLMs provide many advantages over physics-based models and previous probabilistic approaches, ranging from ease of parameter estimation to the ability to incorporate arbitrary features. It is only natural, then, to suspect that these advantages will carry over to related problems as well. For instance, most current methods for multiple sequence RNA secondary structure prediction either take a purely probabilistic approach or attempt to combine physics-based scoring with covariation information in an ad hoc way. In contrast, the CLLM methodology provides a principled framework for combining the rich feature sets of physics-based methods with the predictive power of sequence covariation.

To date, SCFGs and their extensions provide the foundation for many standard computational techniques for RNA analysis, ranging from modeling of specific RNA families to noncoding RNA detection to RNA structural alignment. In each of these cases, CLLMs provide principled alternatives to SCFGs which take advantage of complex features of the input data when making predictions. Extending the CLLM methodology to these cases provides an exciting avenue for future research.

ACKNOWLEDGEMENTS

We thank B. Knudsen for assisting us with Pfold benchmarking, S. R. Eddy and A. Laederach for helpful comments, S. S. Gross and G. Asimenos for helpful discussions regarding algorithms and implementation, and A. F. Novak for assistance in editing the manuscript. CBD was supported by an NDSEG fellowship. Work in the Batzoglou laboratory is supported in part by NSF grant EF-0312459, NIH grant U01-HG003162, the NSF CAREER Award, and the Alfred P. Sloan Fellowship.

REFERENCES

[1] R.D. Dowell and S.R. Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics* 5(71), 2004.

[2] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[3] B. Furtig, C. Richter, J. Wohnert, and H. Schwalbe. NMR spectroscopy of RNA *Chembiochem*., 4(10): 936–962, 2003.

[4] P. P. Gardner and R. Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches *BMC Bioinformatics* 5(140), 2004.

[5] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy. Rfam: an RNA family database. *Nucleic Acids Res*., 31(1):439–441, 2003.

[6] S. Griffiths-Jones, S. Moxon, M. Marshall, A. Khanna, S.R. Eddy, and A. Bateman. Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res.*, 33:D121–D124, 2005.

[7] I.L. Hofacker, W. Fontana, P.F. Stadler, L.S. Bonhoeffer, and P. Schuster. Fast folding and comparison of RNA secondary structures (The Vienna RNA Package). *Monatsh Chem.*, 125:167–188, 1994.

[8] B. Knudsen and J. Hein. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, 15(6): 446–454, 1999.

[9] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res.*, 31(13): 3423–3428, 2003.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th ICML*, pages 282–289, 2001.

[11] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, 288(5):911–940, 1999.

[12] J.S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29: 1105–1119, 1990.

[13] V. Moulton. Tracking down noncoding RNAs. *Proc. Nat Acad. Sci. USA,* 102(7):2269–2270, 2005.

[14] C. Papanicolaou, M. Gouy, and J. Ninio. An energy model that predicts the correct folding of both the tRNA and the 5S RNA molecules. *Nucleic Acids Res.*, 12(1 Pt 1):31–44, 1984.

[15] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing* Cambridge UP, New York, NY, USA, 1992.

[16] J. Reeder, P. Steffen, and R. Giegerich. Effective ambiguity checking in biosequence analysis. *BMC Bioinformatics*, 6(153), 2005.

[17] E. Rivas and S.R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.*, 285:2053–2068, 1999.

[18] E. Rivas and S.R. Eddy. Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics*, 16(7):583–605, 2000.

[19] J.M. Rouillard, M. Zuker, and E. Gulari. OligoArray 2.0: design of oligonucleotide probes for DNA microarrays using a thermodynamic approach. *Nucleic Acids Res.*, 31(12): 3057–3062, 2003.

[20] J. Ruan, G.D. Stormo, and W. Zhang. An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots. *Bioinformatics*, 20(1): 58–66, 2004.

[21] K. Sato and Y. Sakakibara. RNA secondary structural alignment with conditional random fields. *Bioinformatics*, 21(Suppl 2):ii237–ii242, 2005.

[22] I. Tinoco, O.C. Uhlenbeck, and M.D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[23] D.H. Turner, N. Sugimoto, and S.M. Freier. RNA structure prediction. *Ann. Rev. Biophys. Biophys. Chem.,* 17:167–192, 1988.

[24] M.T. Wolfinger, W.A. Svrcek-Seiler, C. Flamm, I.L. Hofacker, and P.F. Stadler. Efficient computation of RNA folding dynamics. *J. Phys. A: Math Gen,* 37: 4731–4741, 2004.

[25] X. Ying, H. Luo, J. Luo, and W. Li. RDfolder: a web server for prediction of RNA secondary structure. *Nucleic Acids Res.,* 32(Web Server Issue):W150–W153, 2004.

[26] M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.,* 31(13):3406–3415, 2003.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e99–e107
doi:10.1093/bioinformatics/btl205

springScape: visualisation of microarray and contextual bioinformatic data using spring embedding and an 'information landscape'

Timothy M. D. Ebbels[1,3], Bernard F. Buxton[2] and David T. Jones[1,*]

[1]Bioinformatics Unit, Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, [2]Department of Computer Science, University College London, Gower Street, London, WC1E 6BT and [3]Current address: Biological Chemistry Section, Biomedical Sciences Division, Imperial College London, Sir Alexander Fleming Building, South Kensington, London SW1 2AZ

ABSTRACT

The interpretation of microarray and other high-throughput data is highly dependent on the biological context of experiments. However, standard analysis packages are poor at simultaneously presenting both the array and related bioinformatic data. We have addressed this challenge by developing a system *springScape* based on 'spring embedding' and an 'information landscape' allowing several related data sources to be dynamically combined while highlighting one particular feature.

Each data source is represented as a network of nodes connected by weighted edges. The networks are combined and embedded in the 2-D plane by spring embedding such that nodes with a high similarity are drawn close together. Complex relationships can be discovered by varying the weight of each data source and observing the dynamic response of the spring network. By modifying Procrustes analysis, we find that the visualizations have an acceptable degree of reproducibility. The 'information landscape' highlights one particular data source, displaying it as a smooth surface whose height is proportional to both the information being viewed and the density of nodes. The algorithm is demonstrated using several microarray data sets in combination with protein-protein interaction data and GO annotations. Among the features revealed are the spatio-temporal profile of gene expression and the identification of GO terms correlated with gene expression and protein interactions. The power of this combined display lies in its interactive feedback and exploitation of human visual pattern recognition. Overall, *springScape* shows promise as a tool for the interpretation of microarray data in the context of relevant bioinformatic information.

Contact: d.jones@cs.ucl.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Microarrays for the measurement of gene expression have become a ubiquitous source of data in many biological experiments. Their strength—the ability to simultaneously track the mRNA levels of thousands of genes—also poses considerable challenges. The high dimensionality and high noise level of the data can obscure patterns that would be recognized with ease in smaller datasets. For these reasons, visualisation of the results of such experiments is difficult and requires sophisticated mathematical tools. Furthermore, the interpretation of experimental results often depends on the biological context, which can be provided by reference to non-microarray data sources. For example, an experiment aimed at distinguishing tumor from normal tissue samples might benefit from a visualisation showing the microarray data mapped to the Gene Ontology (GO) (Ashburner, *et al*., 2000) network of molecular function terms. Alternatively, an experimenter aiming to delineate the transcriptional response to DNA damage might wish to combine microarray data with protein-DNA interaction information in order to highlight genes involved in transcriptional control. However, current analysis packages provide little flexibility to produce single-view visualisations which combine both the gene expression data and complementary information from other bioinformatic resources.

Many bioinformatic data sources can be represented as networks (e.g. protein homology, protein interaction, gene regulation, metabolic networks etc). Even microarray data can be viewed as a network in which genes with similar expression profiles across experimental conditions are connected by links weighted by the strength of the similarity. However, commonly used multivariate visualisation techniques optimize functions which do not explicitly take account of this network structure. For example, principal components analysis (PCA) focuses on the subspace of largest variance within the data, while multidimensional scaling (MDS) optimizes representation of the inter-point distances. One method explicitly designed for visualising network information is that of spring embedding (Eades, 1984; Fruchterman and Reingold, 1991). By representing each node as a mass and each connection as a spring, the method finds a layout of the network which is of low potential energy. In this configuration, nodes connected by the strongest springs (eg genes with the most similar expression profiles) are drawn close to each other while those with weaker interactions lie further apart.

Spring embedding has been used to visualise gene expression data (Schroeder, *et al*., 2001) by representing each gene as a node and using the angle between expression profile vectors as a distance measure. Kim *et al*. (2001) obtained two-dimensional coordinates of

*To whom correspondence should be addressed.

genes from a force-directed placement algorithm (similar to spring embedding) and then summarized the density of points in the third dimension using an 'information landscape'—a 3-dimensional surface whose height is proportional to the density of points in the 2-D plane. This simplification of the complex 2-D layout allowed them to identify regions of the map ('mountains') enriched with genes belonging to specific functional classes. However, in neither of these studies was the important aspect of combining bioinformatic data from multiple sources addressed.

A common approach in microarray analysis is first to explore the gene expression data on its own to ascertain genes and conditions showing interesting behaviour, and later to progress to statistical analyses and integration with functional information. For example, a popular methodology is to generate lists of genes which are differentially expressed between experimental conditions and then to examine them for over- or under-representation of functional annotations, such as those from the GO database. The examination of such lists requires significant effort which increases exponentially with the number of conditions compared. Some of this effort could be reduced if such integration with external information could be achieved at the data-exploratory stage, rather than after explicit hypotheses have been tested. It is toward this goal that our research with spring embedding algorithms has been aimed.

In this paper we describe a system, *springScape*, based on the concepts of spring embedding and the information landscape to visualise gene expression microarray data in the context of other bioinformatic data sources. Our goals are to visualise the high dimensional data in 2-D while combining data from several sources. We desire a method well adapted to visualising biological networks and which enables us to vary the weight of each different data source according to the purpose of the visualisation. The spring embedding technique will allow us to do this in a way which is dynamically visible to the user. Finally, we wish to use the system to focus on one particular aspect of these data sources (often the microarray data itself), and for this we will use the information landscape concept.

We illustrate the use of the *springScape* system with three examples using two microarray data sets. The first example shows how the algorithm can be used to map gene expression data on to a single external data source—a GO network. In the second example, we show how more than one external data source can be combined in a sequential manner to elucidate complex relationships between these and the expression data. Finally, we provide an example showing how correlations between gene expression profiles themselves can be combined with external data to influence the visualisation.

2 METHODS

2.1 Data source combination

Our approach starts by deciding what the N_n nodes and N_e edges in the spring network will represent. For example, each node might be a single gene, with the springs representing the strength of the correlation between the genes' expression profiles over time. Alternatively, each node might be a GO term representing a whole functional class of genes, and the edges could then represent the connectivity of the GO relationships. Next, each of the K information sources must be represented as an N_nxN_n similarity matrix where each element s_{ijk} is a measure of the strength of interaction between nodes i and j in the k'th information source. To continue the examples, in the former case, s_{ijk} might be the thresholded correlation between genes i and j, while in the latter case, s_{ijk} would be the adjacency of terms i and j on the GO directed acyclic graph. When there are several information sources, the individual similarity matrices are combined using a weighted mean:

$$s_{ij} = \frac{1}{K} \sum_k w_k s_{ijk}$$

where w_k is the weight of the k'th information source in the embedding. In our visualisations we used values of w_k ranging from 0.0001 to 0.01. The weights are not normalised so that the absolute magnitude of s_{ij} can be varied by the user.

2.2 Spring embedding

Having defined the overall similarity information to be represented, we proceed to map this to the 2-D plane. We define a network of N_n identical masses connected by N_e springs, where the strength of each spring is specified by the corresponding element of the combined similarity matrix, s_{ij}. If $s_{ij} = 0$ we say the nodes are unconnected, otherwise for both $s_{ij} < 0$ and $s_{ij} > 0$ we say they are connected. The physical nature of the springs is given by a force law which relates the spring length x to the attractive force F. To prevent degenerate solutions (where all nodes collapse to a single point), we ensure that the nodes also repel each other within a small range. The exact form of the force law is not critical to the success of the approach; we follow the approach of Fruchterman and Reingold (1991) in using a square attractive term coupled with an inverse repulsive term. In our system, we subtract a constant repulsive term, r, from all pairs of nodes which are closer than x_{max} to form a modified similarity

$$s'_{ij} = \begin{cases} s_{ij} - r & \text{if } x_{ij} < x_{\max} \\ s_{ij} & \text{if } x_{ij} \geq x_{\max} \end{cases}.$$

Thus nodes separated by less than x_{max} which are unconnected ($s = 0$) receive a negative similarity, while the attraction of connected nodes is reduced. We then compute the spring forces F_{ij} according to

$$F_{ij}(x) = \begin{cases} -s'^2_{ij}(1/x_{ij} - 1/x_{\max}) & \text{if } s'_{ij} < 0 \\ s'_{ij} x^2_{ij} & \text{if } s'_{ij} \geq 0 \end{cases}$$

This set up ensures connected nodes separated by more than x_{max} will always attract and that unconnected nodes coming within a distance x_{max} will always repel each other. However, if two nodes are strongly connected ($s_{ij} > r$) they will always attract even when separated by less than x_{max}. In our experiments, we have found that this helps to unravel 'tangled' network layouts. As the connected nodes are allowed to approach each other closely, they can effectively act as a single node and thus 'thread' their way through gaps that would be inaccessible to a more spatially extended subnetwork. Although this setup theoretically allows two nodes to occupy identical positions, in many biological networks we have analysed, repulsion from the rest of the network prevents this from happening. However, to prevent such pairs collapsing to a single point, we identify connected nodes separated by $x < x_{max}/100$, giving them a repulsive modified similarity of $s'_{ij} = -r$. The values of the parameters used in our visualisations were $x_{max} = 0.2$ and $r = 0.2 \max_k (w_k)$.

The simulation starts with the nodes uniformly distributed at random on the unit square and proceeds according to Newtonian dynamics until the user stops the display (owing to lack of further movement) or a fixed number of iterations has been reached. The equations of motion are solved explicitly using Euler's method. The frequency of display updates is determined by the user (typically every 2–10 iterations) and we used a maximum of between 10^3 and 10^4 iterations in our experiments. (Note that although the basic spring embedding procedure requires around N_n iterations, further iterations are required in the edge crossing and edge repulsion stages—see below). In order to allow the system to reach a static equilibrium, we include a fluid-like energy dissipation term in the simulation. This is defined by a force proportional to, but in the opposite direction to the velocity. The proportionality

constant used in our experiments was 0.1 such that 10% of the velocity is lost at each iteration.

2.3 Edge crossing detection and edge repulsion

An important goal in network layout algorithms is minimisation of the number of edges which cross each other. The obvious advantage is that the structure of a network with fewer edge crossings is much more easily interpretable to the human eye. However, the removal of edge crossings can also lead to a lower energy solution by allowing edges 'stretched' across intervening edges to reduce their length. We include edge crossing detection and correction in our algorithm typically every few hundred iterations of the Euler solver, allowing the system to return to a relaxed state before the next set of corrections are made. Only edges with similarity above a threshold, $s_{ij} > s_{ec}$ are checked and the crossings are corrected using the following algorithm.

Let x_i and y_i, $i = 1,2,3,4$ denote the coordinates of two pairs of nodes in the network connected by two edges (1,2) and (3,4). If $\Delta x_{ij} = x_j - x_i$ and $\Delta y_{ij} = y_j - y_i$ and we calculate

$$\alpha = \frac{\Delta y_{34}\Delta x_{13} - \Delta x_{34}\Delta y_{13}}{\Delta y_{34}\Delta x_{12} - \Delta x_{34}\Delta y_{12}}, \quad \beta = \frac{\alpha\Delta y_{12} - \Delta y_{13}}{\Delta y_{34}},$$

then the two edges cross if $0 < \alpha < 1$ and $0 < \beta < 1$. We correct the crossing simply by swapping the coordinates of points 2 and 4: $(x_2, y_2) \leftrightarrow (x_4, y_4)$. Pairs of edges are considered sequentially and if a crossing is detected, it is immediately corrected. A potential problem here is that a correction may induce new crossings in the remaining network. These are not explicitly checked, but will be corrected if they occur between a pair of edges that has not yet been considered. If not, they will be corrected at the next round of detection.

Although the above algorithm corrects edge crossings, it allows layouts which place some nodes very close to unrelated edges. This makes interpretation of the map difficult because such nodes may appear to be connected to the nearby edge. To reduce this problem we add secondary masses to the centre of each edge which only use the repulsive force law. The 'edge repulsion' is turned on late in the simulation when the large scale layout has been solved and acts merely to obtain a configuration with well spaced nodes.

2.4 Information landscape

Once the 2-dimensional coordinates of the nodes have been found, we are free to use the third dimension to focus on one particular part of the information being presented. We chose to use the idea of an 'information landscape' (Kim, *et al*., 2001)—a 3 dimensional surface whose colour and height represent the information of interest. In our formulation, we construct the surface using Gaussian kernels of fixed width σ placed at each node. The height z of the surface at any point (x,y) is found by summing the kernels:

$$z(x,y) = \sum_{i=1}^{N} u_i \exp\left(-\frac{(x-x_i)^2 + (y-y_i)^2}{2\sigma^2}\right)$$

where each kernel is weighted by u_i, the parameter of interest (eg mean gene expression level for node i). The kernel width can be set by the user according to the level of detail they wish to display. The figures in this paper used values between $\sigma = 0.01$ and 0.02.

2.5 Assessing reproducibility of the visualisation

Given the same data, the spring embedding approach does not always produce the same layout, partly owing to the random initial configuration and partly because of the presence of local minima in the spring potential energy. We assessed how often a similar layout is obtained over many random restarts using two methods. Firstly, we developed a modified form of Procrustes analysis (see supplementary information) to examine the similarity between different layouts. The analysis was applied to subnetworks of varying sizes and the evolution of Procrustes fit error with subnetwork size was compared to that from random layouts. Secondly, we investigated the potential energies of the spring layouts. The mean and standard deviation of the energy was calculated across the multiple realisations and compared to that from random layouts (see Supplementary Information).

springScape and all testing procedures were developed within the MATLAB programming environment (version 7.0, The MathWorks, Natick, MA) on a 2.5GHz Pentium 4 PC with 1GB RAM. We use MATLAB in preference to other data analysis tools such as R because of its highly interactive graphics facilities.

3 RESULTS

3.1 Yeast cell cycle time course displayed on GO

To demonstrate the visualisation of gene expression data in the context of a single additional data source, we used the yeast cell cycle data of Spellman *et al*. (1998). In this experiment, yeast cells were synchronised by the addition of α factor, and the cell population sampled every 7 minutes over the course of approximately two cell division cycles (119 min in total). RNA was extracted and expression profiles obtained using spotted cDNA arrays. We mapped the genes to the GO cellular component ontology using annotations from the Saccharomyces Genome Database (SGD) (Cherry, *et al*., 1998). A subnetwork of the GO graph, rooted at the term 'cell' and extending to a distance of 2 edges was extracted and a visualisation using the spring embedding algorithm is shown in Figure 1. The layout clearly illustrates the GO connectivity and satisfies various 'aesthetic' criteria such as a low number of edge crossings.

The mean expression ratio (MER) of all genes annotated to each GO term was then visualised using the information landscape technique. To calculate the MER for the leaf nodes of the subnetwork, we used the inheritance properties of GO to allocate genes to leaves that were ancestors of the annotated terms. However, inheritance was *not* used at higher levels of the subnetwork itself, to avoid the MER of an ancestor being influenced by genes annotated to a descendant present in the visualisation.

The data for 4 of the time points are shown in Figure 2. The GO network has been visualised by the spring embedding procedure and is overlaid by a landscape of mean gene expression. Some nodes have no annotations for yeast and therefore display no landscape in the figure. Clear differences between the mean expression of each GO term can easily be identified from this plot. For example, at 0 min, the term with the highest mean expression, to the right of the network, is 'cell wall'. High expression of cell wall proteins might be expected at mitosis which is the stage depicted at the initiation of the experiment. As the cells move into the G1 and S phases of the cell cycle at 14 and 28 min, mean expression becomes more comparable across many GO terms, particularly children of the terms 'intracellular' and 'membrane'. This includes, for example, the term 'nucleus' and terms relating to the mitochondrial membrane and respiratory chain complexes. At 63 min, the cells are beginning the second round of the cycle, again in the M/G1 phase. However, the plot indicates that the gene expression profile is far from similar to that at 0 min, being heavily dominated by the term 'septum'. Again, activity at this location is expected at mitosis and there is one gene in the GO database, DSE4 (YNR067C), annotated to this term. This gene is involved in degrading the cell wall, allowing mother and daughter cells to

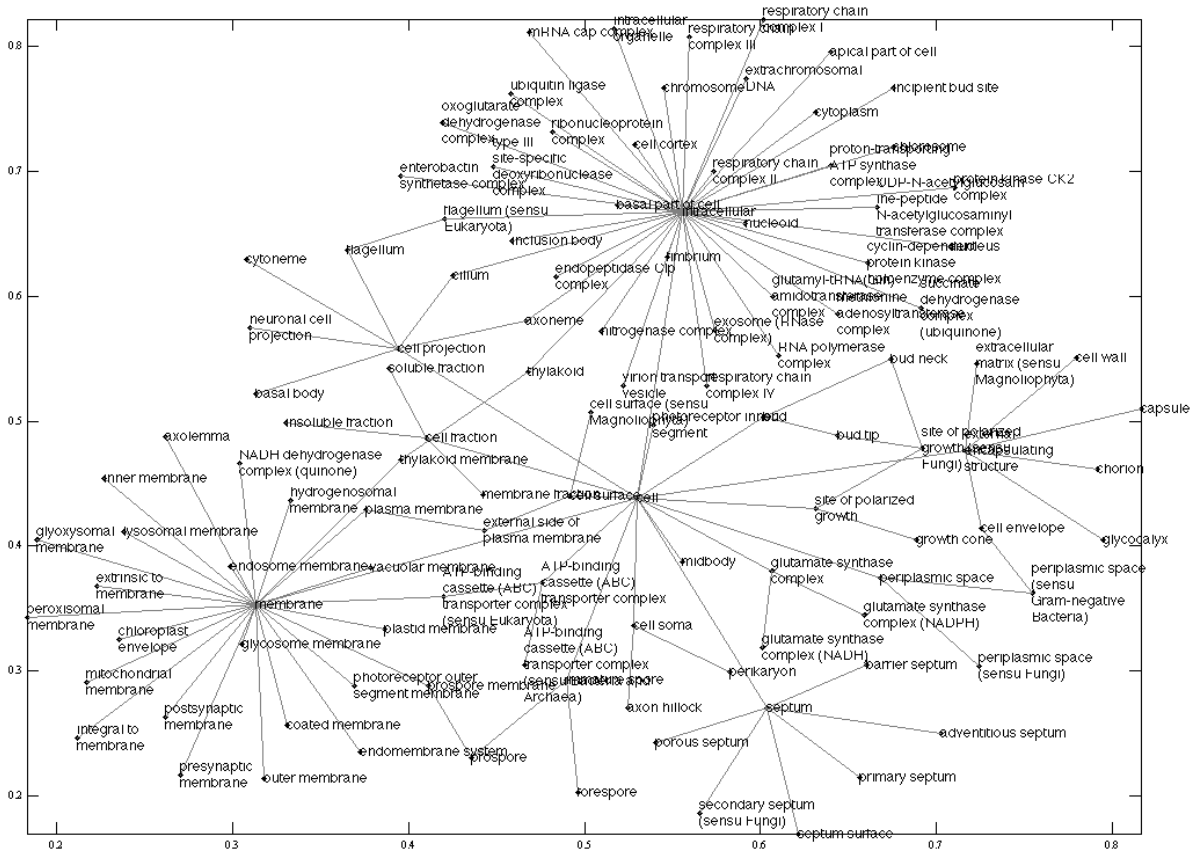


Fig. 1. Part of the GO cellular component ontology visualised by the spring embedding algorithm. The ontology is rooted at the term 'cell' and extends to a distance of 2 edges. The directions of the GO relationships are not shown to avoid complicating the display.

separate (Colman-Lerner, *et al*., 2001). Note that GO terms are highlighted by MER, not by enrichment in differentially expressed genes, and so the standard hypergeometric test for enrichment is not applicable. Thus, although the mean is a rather course summary of the expression of many genes annotated to a given term, this is not always a limiting factor and the system affords a global overview of the expression changes in the context of the Gene Ontology relationships. Changes in expression related to different locations in the cell can be easily identified, and interesting regions expanded by extracting further subnetworks of GO at a deeper level.

3.2 Yeast cell cycle displayed on GO with protein–protein interactions

The real versatility of the system becomes apparent when gene expression data is viewed in the context of more than one other information source. For yeast, a wealth of bioinformatic data is available, much of which can be represented as a network. We chose data on protein-protein interactions (PPIs) from the Munich Information centre for Protein Sequences (MIPS, http://mips.gsf.de/) to add to the visualisation described above. Figure 3 shows the effect of adding this as a second external information source. The upper left panel depicts the GO cellular component ontology relationships, rooted at the term 'cell' and visualised by the spring embedding procedure (as for Figure 2). In the upper right panel, PPIs have been added to the information displayed. Note that since GO inheritance is not used except at the leaf nodes, none of the PPIs shown connect ancestor - descendant pairs. The number of interacting protein pairs for each pair of GO terms gives an interaction strength which is depicted by the thickness of the red lines in the figure. In the lower left panel, the PPI springs have been activated, leading to a new network layout where GO terms which share many interactions are drawn closer together. To highlight terms most affected by the addition of this PPI information, pairs of terms were ranked by the magnitude of the decrease in separation due to the new forces. The pairs showing the five largest decrements in separation are all between children of the terms 'membrane' and 'intracellular'. For example, the largest change is for the terms 'integral to membrane' and 'cytoplasm' which have 523 distinct pairs of interacting proteins linking them.

Finally, the lower right panel shows, for the 28 min time point, the mean expression of all genes annotated to each GO term as a coloured landscape. This is superimposed on the network layout produced by both the GO and PPI information. This time point is the same as that shown in the lower left panel of Figure 2, but Figure 3 gives a very different picture of the landscape of gene expression. In Figure 2, the distribution of mean expression is rather uniform across GO terms whereas the landscape of Figure 3 is dominated by a few salient peaks. These high peaks highlight cellular components characterised by both large numbers of protein interactions and high mean gene expression.

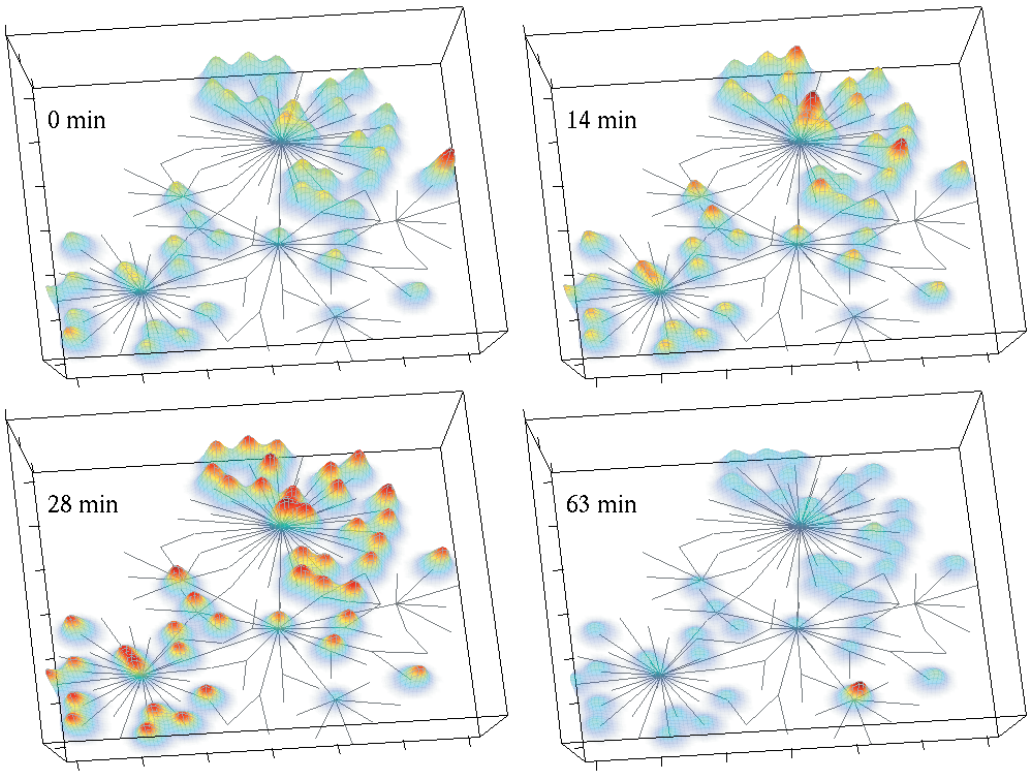


Fig. 2. Yeast cell cycle data. The GO cellular component ontology of Figure 1 is shown (solid lines) with the mean expression of all genes annotated to each node overlaid in the 3rd dimension (coloured landscape). Each panel shows expression data for a different time point in the cell cycle. Note that the height and colour of the landscape in each panel has been scaled to maximise differences in mean expression between the nodes, and thus is not comparable between panels.

The highest peak towards the top middle of the panel is due to the terms 'cytoplasm', 'intracellular organelle' and 'nucleus'. The number of interacting protein pairs between each of these three terms is greater than 3400 and has led to their close proximity on the map. Note that the three terms corresponding to this peak are terminal nodes of the network (i.e. at the lowest level displayed in Figure 1).

3.3 Human dendritic cell viral infection data

Our last example is aimed at demonstrating how the visualisation can be influenced by genes with similar patterns of expression across a time course or multiple experimental conditions. Human dendritic cells were exposed to five stimuli, either pathogen components (lipopolysaccharide (LPS), polyinosinic polycytidylic acid (PIC, a synthetic form of double-stranded RNA)), or live and inactivated viruses (influenza, UV-treated influenza, rhinovirus). Expression profiles were monitored at 6 time points over 24 hours using custom cDNA arrays. The top level network of the GO molecular function ontology was extracted from the GO database extending to all terms with more than 300 UniProt annotations (http://www.ebi.uniprot.org/). Genes from the custom arrays were assigned to the GO terms using Unigene (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene and Locus Link (www.ncbi.nlm.nih.gov/LocusLink) mappings. The network as visualised by the spring embedding algorithm is shown in Figure 4.

Figure 5 (upper panel) shows the mean log expression ratio for the LPS treatment at 18 hours of all genes annotated to each term visualised by the information landscape technique. As with the yeast data, GO inheritance was only used to annotate the leaf nodes and not the internal nodes of the network. The plot is dominated by two peaks—one positive and one negative. The positive peak corresponds to the term 'rhodopsin-like receptor activity' and the negative one to 'protein kinase activity'. Next we add springs whose strengths are proportional to the Pearson correlation coefficient between the mean expression profiles (across all time points and experimental conditions) of each pair of terms. GO terms whose mean expression responds over time and condition in a similar fashion to each other will thus be drawn closer together on the resulting map. This is shown in Figure 5 (lower panel) where we see that the GO network has become distorted by these extra forces. Disregarding terms which have a parent–child relationship, the pair having the highest correlation are 'carrier activity' and 'cation transporter activity' which, annotate identical genes in this data set and therefore have a correlation of 1.0. This pair of nodes is found close together at approximately (0.4,0.4) in the lower panel. Other pairs of terms found close together in the lower panel include 'endopeptidase activity' and 'cation channel activity' at approximate coordinates (0.45,0.53) with correlation 0.62. If one plots the inter-node distances from the lower panel versus the correlation coefficients, it is found that there is a

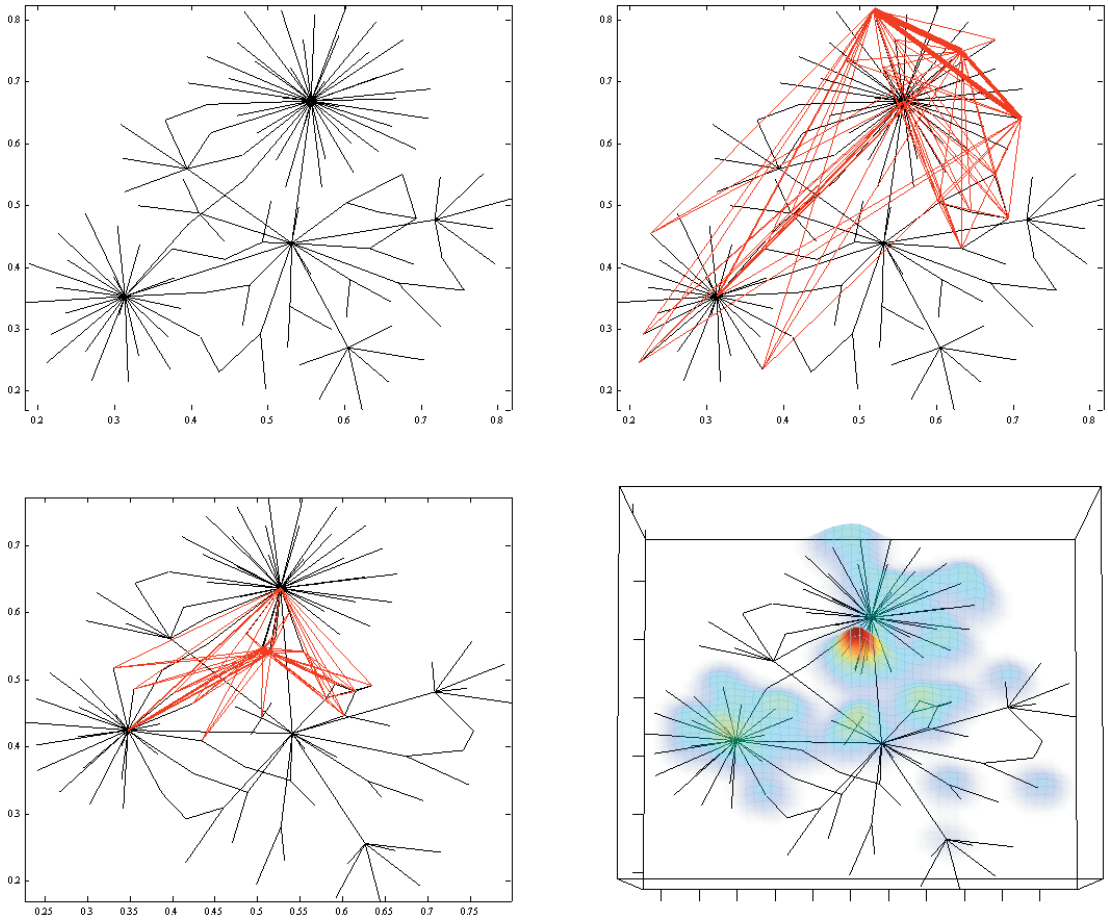


Fig. 3. The GO cellular component ontology network (solid lines) is shown (upper left) rooted at the term 'cell'. Protein-protein interactions are added (upper right) in red, where the line thickness is proportional to the number of interactions between each pair of GO terms. Only pairs with the largest 10% of interactions are shown for clarity. The springs corresponding to the interactions are activated and the network re-visualised (lower left), resulting in a new landscape of mean gene expression for the 28 min time point (lower right).

relationship (though weak, Pearson correlation −0.55) such that pairs with higher correlations tend to have lower inter-node distances (data not shown). Of course, one does not expect a perfect relationship since the visualisation is not merely optimising the correlation springs, but also those of the GO network.

3.4 Reproducibility of the visualisation

Though the Newtonian dynamics of the spring algorithm are deterministic, repeated layouts of the same input data may not be identical due to the random initial configuration and existence of local minima in the spring energy. We assessed the reproducibility of the spring embedding algorithm for two different biological networks: the top level of the GO biological process ontology (down to 300 UniProt annotations) and part of the MIPS PPI network for yeast (rooted at YAL003w and extending to a distance of 5 edges).

As expected, the reproducibility decreased with increasing scale (see Table 1 in Supplementary Information), with the Procrustes residual increasing from 10–15% at small scales to 25–30% at scales equal to half the network size. The residuals for the PPI network were somewhat higher than those for the GO network. We attribute this to the higher connectivity of the PPI network (mean connection degree 2.9 as compared to 2.4 for the GO network). This increases network 'frustration', thereby multiplying the number of similar local minima of the spring potential energy. Comparison with random layouts shows that the spring-embedded layouts achieve a much lower residual, around 15–30% of the random value. In all cases the reproducibility of the real layouts was much higher than random layouts.

Additionally, we compared the mean potential energy of the springs in each layout to assess the similarity of the (possibly local) minima found by the multiple realisations. For the GO network the mean energy per spring was 2.50±0.010 while for the PPI network it was 2.43 ± 0.003 (arbitrary units). The standard errors demonstrate a low degree of variation in energy minima across the multiple embeddings. The mean energies of the random layouts were 799 ± 9.7 and 757 ± 5.0 respectively, several hundred times larger than for the embedded layouts, again showing the substantial improvement produced by the embedding algorithm. In summary therefore, we find that while the layouts produced by the spring embedding algorithm are not always identical, there is a reasonable degree of reproducibility which is several

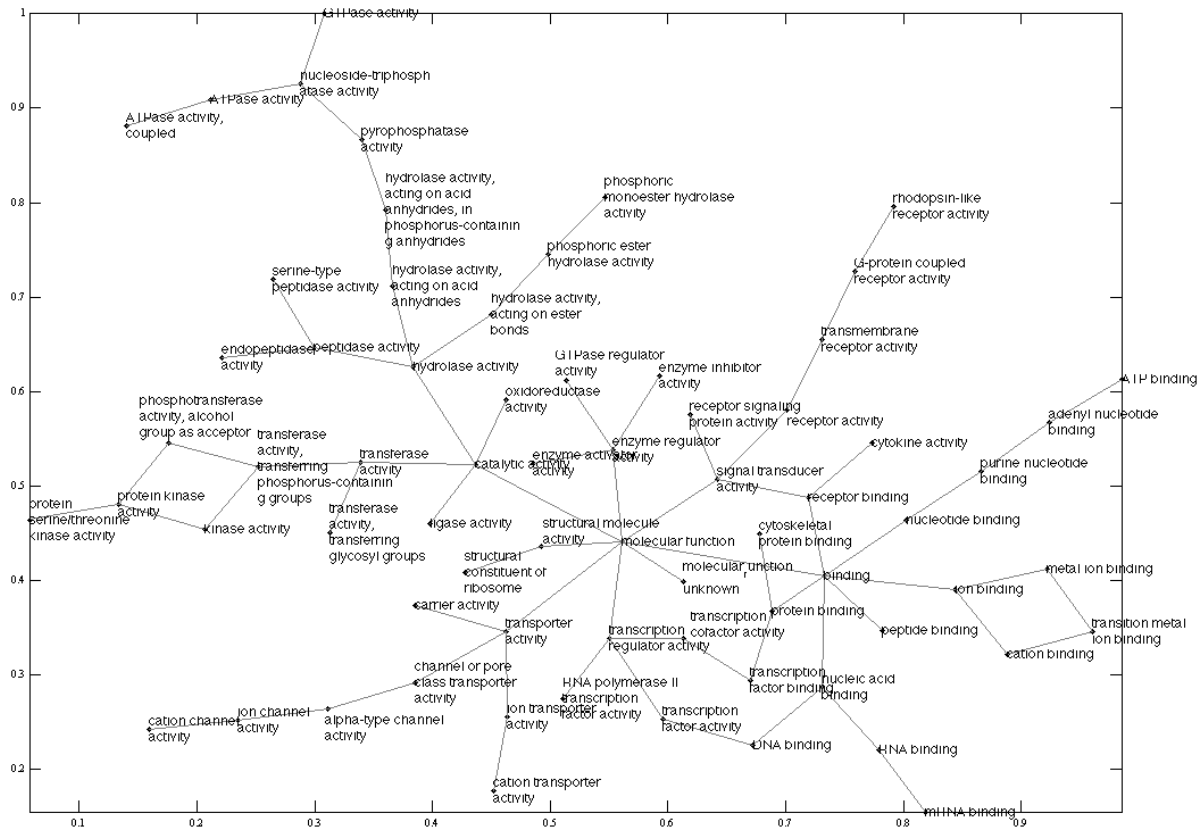


Fig. 4. The top level of the GO molecular function ontology as visualised by the spring embedding algorithm. The ontology is rooted at the term 'molecular function' (node 3) and extends to all terms with more than 300 UniProt annotations. Arrows showing the direction of GO relationships have been omitted for clarity.

times better than that for random layouts. Although it is difficult to translate these layout reproducibility values into the reproducibility of interpretation, experience with a number of different networks indicates that the variation is not usually high enough to change the overall inferences drawn from the visualisation.

4 DISCUSSION

The system developed here is motivated by our desire to combine several bioinformatic information sources when visualising the results of a gene expression microarray experiment. It is often through combination of different experimental data that scientific insight is generated and this is made easier if all relevant information sources are synthesised in a single view. The representation of the data as networks allows the simultaneous combination of many different sources of information and the solution described is flexible in that the influence of each information source can be varied by the user. Typically this would involve the sequential 'turning on' of forces for each set of data and observation of the dynamic response of the spring network, as exemplified by Figure 3. Alternatively, one could alternate back and forth between visualisations based on different information sources in order to gauge the influence of each one. The fusion of multiple data sources is augmented by adding a third dimension displaying one key aspect of the data, often the gene expression levels themselves. With the information landscape technique, the display of this 'special' information is influenced by the spring-embedded map; the height of the landscape is a combination of the values of the 'special' data and the density of nodes. We have shown that the system is capable of highlighting known biological information (such as the high mean expression of cell wall and septum proteins in Figure 2). The potential for identification of novel biology by combining different information sources is exemplified by the GO / gene expression / PPI visualisation of Figure 3. Nonetheless it should be stressed that such visualisations merely serve to generate hypotheses which must then be confirmed by further statistical or experimental investigation.

Though we have illustrated the use of *spingScape* primarily with examples from GO, we emphasise that this is *not* just another GO browser. There already exist many tools for visualising the enrichment of GO terms with differentially expressed genes (e.g. Zeeberg, *et al*., 2003; Zhang, *et al*., 2004). In our case, GO simply provides one particularly useful bioinformatic context to influence the visualisation. For the display of biological networks in general, several methods have been proposed (Enright and Ouzounis, 2001; Kim, *et al*., 2001; Schroeder, *et al*., 2001; Han and Ju, 2003; Adai, *et al*., 2004). However, none of these show how information sources can be combined to investigate their different influences on the interpretation. We have chosen to combine networks through a simple weighted mean of similarity matrices, though more complex ideas can be envisaged, such as making use of networks with different

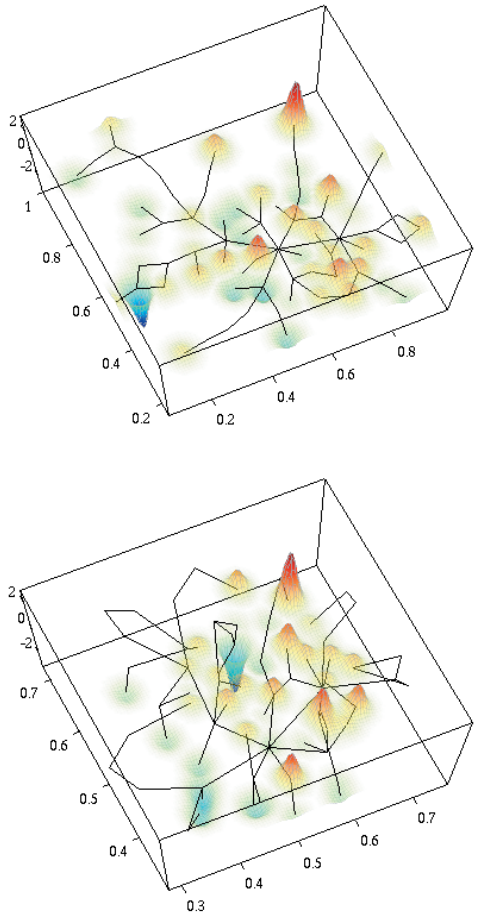


Fig. 5. Human dendritic cell data. upper panel: the GO molecular function network of Figure 4 (solid lines) overlaid with a landscape showing the mean expression (log ratio) of all genes annotated to each term. The landscape corresponds to the LPS treatment at 18 hours. In the lower panel, springs corresponding to the correlations between the expression profiles of each node are turned on and the network and landscape are revisualised.

edge types. For example, we have demonstrated our method with undirected networks (e.g. GO was treated as undirected), but directed representations could also be considered. These correspond to non-symmetric similarity matrices and when combined, could cause effects such as the cancelling out of opposite-direction edges and the reinforcement of same-direction edges.

Our spring embedding algorithm is designed to visualise networks with up to a few hundred nodes and edges. At the outset, we realised that attempting to visualise larger networks would be fruitless, since even for 'aesthetically pleasing' layouts, the display space would become cluttered with nodes and edges. The time complexity of the algorithm depends on the range of the repulsive force, x_{max}. When this is large, most nodes repel each other and the time required for each iteration is approximately quadratic in the number of nodes, $t \propto N_n^2$. When x_{max} is small, the iteration time depends linearly on the number of edges in the network, $t \propto N_e$. Since in a connected network N_e can vary between $N_n - 1$ and $\frac{1}{2}N_n(N_n - 1)$, in most practical applications, each iteration proceeds at quadratic speeds. When combined with the total number of iterations, this produced an approximately cubic total time complexity in our experiments. This rather strong scaling with the number of nodes is not a disadvantage however, since we do not plan to visualise very large networks for the reasons noted above. Other algorithms such as LGL (Adai *et al*., 2004) and Interviewer (Han and Ju, 2003) are specifically aimed at scalability, and are more appropriate for straightforward layout of large graphs. In practice, the examples presented here were all visualised in less than 2 minutes on a 2.5GHz PC with 1GB RAM running MATLAB 7.0. In addition, we have found that further insight into the relationships within the data is obtained by observing the *dynamics* of the algorithm, rather than merely viewing the final layout. For example, the strength of a connection is perceived intuitively from the acceleration of the connected nodes—a feature not explicitly visible in the final layout. Finally, few algorithms allow one data source to receive a special focus as provided by the information landscape. To our knowledge, this concept has only been used to represent the clustering of nodes in the layout (Kim, *et al*., 2001), rather than a combination of the node density and other information of particular interest, as presented here. For display, the landscape is evaluated on a grid, and its display scales with the grid size (not the network size) with typical display times of $\sim$10s using the equipment described above.

We have demonstrated *springScape* on a few examples, specifically concentrating on networks derived from the Gene Ontology since these are some of the most common annotation resources used in the interpretation of microarray experiments. However, many applications can be envisaged, such as combining gene regulatory interactions from the literature with time-series correlations between the expressed genes. This would highlight when existing interactions are supported by the experimental data, and allow new interactions to be hypothesised. Alternatively, the similarities between expressions of a group of genes could be used to embed a network where the nodes represent arrays. This would form a map showing differences between treatment conditions, in a similar fashion to that in which PCA is often used. However, with the algorithm described here, further information linking the arrays (perhaps clinical chemistry or tumour morphology measurements) could be included in the display. As shown in this paper, the Gene Ontology perhaps provides some of the most interesting applications. For example, combining a GO network with protein homology information, one could construct maps similar to those presented here, but where the GO network is distorted in a fashion specific to a given organism or group of organisms.

5 CONCLUSIONS

In summary, the *springScape* system presented here allows the results of microarray experiments to be viewed in their biological context; in particular, several relevant bioinformatic data sources may be combined to produce a visualisation which reflects the biological question being addressed. This type of combined visualisation is not present in currently available algorithms for the display of such data, leaving the biologist struggling to see the 'big picture' and the overall meaning of the experimental results. We hope that our system may be used in the future to help generate clarity and insight from the synthesis of microarray data with multiple other sources of biological information.

ACKNOWLEDGEMENTS

The authors acknowledge Dr. Paul Kellam and Antonia Kwan for providing the dendritic cell data. TMDE acknowledges The Wellcome Trust Biomap project for financial support.

REFERENCES

Adai,A.T., Date,S.V., Wieland,S. and Marcotte,E.M. (2004) LGL: creating a map of protein function with an algorithm for visualizing very large biological networks, J. Mol. Biol., 340, 179.

Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T., Harris,M.A., Hill,D.P., Issel-Tarver,L., Kasarskis,A., Lewis,S., Matese,J.C., Richardson,J.E., Ringwald,M., Rubin,G.M. and Sherlock,G. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, Nat Genet, 25, 25–29.

Cherry,J.M., Adler,C., Ball,C., Chervitz,S.A., Dwight,S.S., Hester,E.T., Jia,Y., Juvik,G., Roe,T., Schroeder,M., Weng,S. and Botstein,D. (1998) SGD: Saccharomyces Genome Database, Nucleic Acids Res., 26, 73.

Colman-Lerner,A., Chin,T.E. and Brent,R. (2001) Yeast Cbk1 and Mob2 activate daughter-specific genetic programs to induce asymmetric cell fates. Cell, 107, 739–750.

Eades,P. (1984) A heuristic for graph drawing, Congressus Numerantium, 42, 149.

Enright,A.J. and Ouzounis,C.A. (2001) BioLayout—an automatic graph layout algorithm for similarity visualization, Bioinformatics, 17, 853.

Fruchterman,T.M.J. and Reingold,E.M. (1991) Graph Drawing by Force-directed Placement, Software—Practice and Experience. 21, 1129.

Han,K. and Ju,B.H. (2003) A fast layout algorithm for protein interaction networks, Bioinformatics, 19, 1882.

Kim,S.K., Lund,J., Kiraly,M., Duke,K., Jiang,M., Stuart,J.M., Eizinger,A., Wylie,B.N. and Davidson,G.S. (2001) A gene expression map for Caenorhabditis elegans, Science, 293, 2087.

Schroeder,M., Gilbert,D., van Helden,J. and Noy,P. (2001) Approaches to visualisation in bioinformatics: from dendrograms to Space Explorer, Information Sciences, 139, 19.

Spellman,P.T., Sherlock,G., Zhang,M.Q., Iyer,V.R., Anders,K., Eisen,M.B., Brown,P.O., Botstein,D. and Futcher,B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization, Mol. Biol. Cell, 9, 3273.

Zeeberg,B.R., Feng,W., Wang,G., Wang,M.D., Fojo,A.T., Sunshine,M., Narasimhan,S., Kane,D.W., Reinhold,W.C., Lababidi,S., Bussey,K.J., Riss,J., Barrett,J.C. and Weinstein,J.N. (2003) GoMiner: a resource for biological interpretation of genomic and proteomic data, Genome Biol, 4, R28.

Zhang,B., Schmoyer,D., Kirov,S. and Snoddy,J. (2004) GOTree Machine (GOTM): a web-based platform for interpreting sets of interesting genes using Gene Ontology hierarchies, BMC Bioinformatics, 5, 16.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e108–e116
doi:10.1093/bioinformatics/btl231

Analysis of sample set enrichment scores: assaying the enrichment of sets of genes for individual samples in genome-wide expression profiles

Elena Edelman[1,2], Alessandro Porrello[1,3,4,7], Justin Guinney[1,2], Bala Balakumaran[1,3,4], Andrea Bild[1,3], Phillip G. Febbo[1,3,4] and Sayan Mukherjee[1,5,6,*]

[1]Institute for Genome Sciences & Policy, [2]Computational Biology and Bioinformatics Program, [3]Department of Medicine, [4]Department of Molecular Genetics and Microbiology, [5]Institute of Statistics and Decision Sciences, [6]Department of Computer Science, Duke University, Durham NC 27708, USA and [7]Molecular Oncogenesis Laboratory, Regina Elena Cancer Institute, Via Delle Messi D'Oro 156, Rome, 00158, Italy

ABSTRACT

Motivation: Gene expression profiling experiments in cell lines and animal models characterized by specific genetic or molecular perturbations have yielded sets of genes annotated by the perturbation. These gene sets can serve as a reference base for interrogating other expression datasets. For example, a new dataset in which a specific pathway gene set appears to be enriched, in terms of multiple genes in that set evidencing expression changes, can then be annotated by that reference pathway. We introduce in this paper a formal statistical method to measure the enrichment of each sample in an expression dataset. This allows us to assay the natural variation of pathway activity in observed gene expression data sets from clinical cancer and other studies.

Results: Validation of the method and illustrations of biological insights gleaned are demonstrated on cell line data, mouse models, and cancer-related datasets. Using oncogenic pathway signatures, we show that gene sets built from a model system are indeed enriched in the model system. We employ ASSESS for the use of molecular classification by pathways. This provides an accurate classifier that can be interpreted at the level of pathways instead of individual genes. Finally, ASSESS can be used for cross-platform expression models where data on the same type of cancer are integrated over different platforms into a space of enrichment scores.

Availability: Versions are available in Octave and Java (with a graphical user interface). Software can be downloaded at http://people.genome.duke.edu/assess

Contact: sayan@stat.duke.edu

1 INTRODUCTION

Gene expression profiling experiments have been conducted on a wide variety of cell lines and animal models with the goal of characterizing genes sets whose expression patterns characterize specific genetic or molecular perturbations. These gene sets contain candidate players in pathways, or sub-pathways, that are ''annotated'' by the experimental perturbation. The fundamental idea in pathway based analysis approaches (Huang *et al*., 2003; Black *et al*., 2003; Mootha *et al*., 2003; Sweet-Cordero *et al*., 2005; Alvarez *et al*., 2005; Febbo *et al*., 2005; Subramanian *et al*., 2005) is that such a gene set serves as a reference base for interrogating other expression data sets. A new data set in which a specific pathway gene set appears to be enriched, in terms of multiple genes in that set evidencing expression changes can then be annotated by that reference pathway. An analogy can be made here with sequence annotation in a BLAST search: sets of experimentally derived pathways serve as annotation reference sets for future experiments in the same way that annotated sequences serve as references in a sequence search. Statistical methods are needed and have been developed (Subramanian *et al*., 2005; Barry *et al*., 2005; Kim and Volsky, 2005; Tomfohr *et al*., 2005) to define computational tools for such expression-based pathway annotation. Two of these methods, GSEA (Subramanian *et al*., 2005) and SAFE (Barry *et al*., 2005), use nonparametric statistics to provide formal statistical evaluation, and confidence assessments, for annotation of an expression data set by measuring the overlap of significantly perturbed genes with those in each pathway in a database of pathways. Gene Set Enrichment Analysis (GSEA) (Subramanian *et al*., 2005) has been successfully applied in many basic science and clinical studies (Mootha *et al*., 2003; Sweet-Cordero *et al*., 2005; Alvarez *et al*., 2005; Febbo *et al*., 2005; Subramanian *et al*., 2005; Bild *et al*., 2006), including pathway deregulation in cancer genomics. A fundamental shortcoming of GSEA and other methods (Barry *et al*., 2005; Kim and Volsky, 2005; Tomfohr *et al*., 2005) is that they do not characterize the variation in enrichment over individual samples in the data set.

If the enrichment of each sample in an expression data set can be annotated, then one can assay the natural variation of pathway activity in observed gene expression data sets from clinical cancer and other studies. The ability to assay pathway variation in samples allows the implementation of a general methodology to dissect tumor samples in terms of oncogenic pathways. The logic behind this methodology is to develop gene expression ''signatures'' of

*To whom correspondence should be addressed.

oncogenic pathways from model systems and then use these ''signatures'' to annotate human tumors in terms of the deregulation of oncogenic pathways (Bild *et al*., 2006).

In this paper we introduce a statistical method that allows us to assay pathway variation, Analysis of Sample Set Enrichment Scores (ASSESS). Given gene sets defined by prior biological knowledge or genes co-expressed in an experiment with a specific genetic or molecular perturbation, and a data set of expression profiles from samples belonging to two classes, ASSESS provides: a measure of the enrichment of each gene set in each sample and a confidence assessment. This extends the methodology developed in GSEA and SAFE to annotate individual samples.

A family of methods for pathway annotation was developed and used to measure pathway deregulation in breast cancer and lung cancer (Huang *et al*., 2003; Black *et al*., 2003; Bild *et al*., 2006). The approach involved: (a) building statistical models of pathway deregulation from cell lines where recombinant adenoviruses were used to express oncogenic activities corresponding to pathway deregulation, (b) applying the models to each sample in a data set of tumors and estimating the probability of deregulation of the pathways. The main drawback of this methodology is that cell line perturbation data as well as tumor data are required for the analysis. For ASSESS, only the list of genes characterizing the pathway deregulation is required, the entire model and cell line data is not needed. This provides a great advantage when the gene sets are determined by literature review or a non-expression based assay, such as immunohistochemical, for which building an accurate model subsequently applicable to expression data is a very difficult challenge.

2 RESULTS

2.1. Analysis of sample set enrichment scores

ASSESS is an annotation methodology that takes as inputs:

(1) Genome-wide expression profiles consisting of p genes and n samples with each sample corresponding to one of two classes, C_1, C_2. The expression of the j-th gene in the i-th sample is x_j^i;

(2) A database of m gene sets $\Gamma = \{\gamma_1, \ldots \gamma_m\}$ where each gene set γ_k is a list of genes (a subset of the p genes in the data set) belonging to a pathway or other functional or structural category;

(3) A ranking procedure and correlation statistic that takes the expression data set and labels as inputs and produces correlation statistics for each sample that reflects the correlation of the p genes in that sample with respect to the the distribution of expression in the two classes. The correlation statistics for the i-th sample would be $\mathbf{c}_i = \{c_1^i, \ldots, c_p^i\}$ where c_j^i is calculated by any likelihood ratio statistic for measuring the correlation of a sample to one class rather than the other:

$$c_j^i = \log\left(\frac{\mathbb{P}(x_j^i \in C_1 \,|\, x_j^i, \text{data})}{\mathbb{P}(x_j^i \in C_2 \,|\, x_j^i, \text{data})}\right).$$

An example of a parametric and nonparametric correlation statistic is described in detail in the following sections;

and produces as outputs:

(1) An enrichment score for each sample in the data set with respect to each gene set in the database. ES_i^k corresponds to the enrichment of the i-th sample with respect to the k-th gene set;

(2) A measure of confidence for each enrichment score is given by a p-value with multiplicity taken into account by Family-wise error rate (FWER) p-values and False Discovery rate (FDR) q-values.

Given the correlation statistics for the i-th sample, $\mathbf{c}_i = \{c_1^i, \ldots, c_p^i\}$, and a gene set γ_k, we construct the following discrete random walk over the indices of the rank-ordered correlation statistic

$$\nu(\ell) = \frac{\sum_{j=1}^{\ell} |c_{(j)}|^{\tau} I(g_{(j)} \in \gamma_k)}{\sum_{j=1}^{p} |c_{(j)}|^{\tau} I(g_{(j)} \in \gamma_k)} - \frac{\sum_{j=1}^{\ell} I(g_{(j)} \notin \gamma_k)}{p - |\gamma_k|}, \qquad (1)$$

where $c_{(j)}$ is the rank-ordered correlation statistic, τ is a parameter (in general $\tau = 1$), γ_k is the k-th gene set, $I(g_{(j)} \in \gamma_k)$ is the indicator function on whether the j-th gene (the gene corresponding to the j-th ranked correlation statistic) is in gene set γ_k, $|\gamma_k|$ is the number of genes in the k-th gene set, and p is the number of genes in the data set. The enrichment statistic for the i-th sample with respect to the k-th gene set is the maximum deviation of the random walk from zero

$$ES_i^k = \nu[\arg\max_{\ell=1,\ldots,p} |\nu(\ell)|]. \qquad (2)$$

The random walk is a tied-down Brownian bridge process and the deviation from zero is very closely related to the classical Kolmogorov-Smirnov statistic (Feller, 1971). There are simpler ways to define the enrichment score such as taking the average rank of the genes in the gene set from the rank-ordered list of genes. However, using a random walk is advantageous because it allows one to see how the genes in the set are distributed in the rank-ordered list. The random walk could alternatively be solved by ranking $|c_{(j)}|$ rather than $c_{(j)}$. While in certain situations a more extreme enrichment score may be sacrificed, this ranking will not allow for the access to the additional information of which genes in the set are up or down regulated. Therefore we choose to calculate the ES by a random walk using values ranked by $c_{(j)}$, and suggest gene sets be constructed to capture genes correlated with either over expression or under expression in a class but not both.

To measure significance we assume under the null hypothesis that the labels are exchangeable and therefore we can compute the null distribution by permuting labels, ranking the genes according to the recomputed statistic $c_{(j)}^{\pi}$, and computing the ''random'' enrichment statistic $ES_i^k(\pi)$. This is done over many label permutations, $\pi = 1, \ldots, \Pi$. The p-value is computed by comparing the enrichment score to the empirical distribution generated from $\{ES_i^k(\pi)\}_{\pi=1}^{\Pi}$. Correction for multiple hypothesis testing is preformed in the same manner as in GSEA and is addressed via FWER p-values or FDR q-values (see (Subramanian *et al*., 2005) for details). Overlapping gene sets do not influence the calculation of q-values.

The key technical innovation in extending methods such as GSEA or SAFE to provide enrichment scores for individual samples is producing a correlation statistic and subsequent rankings that model how representative each gene for a given sample is with respect to the two classes. The ranking should reflect the natural variation of how each sample is correlated with class labels. We introduce two correlation statistics which reflect this variation: (1)

based on a simple parametric normal model, (2) based on a nonparametric random walk model. Both methods preform well, as likely would other well-defined likelihood ratios such as a binary regression model. We choose to use the nonparametric model on the data in this paper as this method is a novel means for calculating class membership likelihoods. However, the parametric method as well as other methods not discussed here give comparable results.

2.1.1 Parametric model The parametric model assumes that the expression of a given gene can be modeled by a mixture of two normal distributions corresponding to the two classes. The mean and standard deviations are computed from the data

$$\hat{\mu}_{j1} = \frac{1}{n_1}\sum_{i\in C_1} x_j^i, \quad \hat{\mu}_{j2} = \frac{1}{n_2}\sum_{i\in C_2} x_j^i,$$
$$\hat{\sigma}_{j1}^2 = \frac{1}{n_1}\sum_{i\in C_1} (x_j^i - \hat{\mu}_{j1})^2, \quad \hat{\sigma}_{j2}^2 = \frac{1}{n_2}\sum_{i\in C_2} (x_j^i - \hat{\mu}_{j2})^2,$$

where n_1 and n_2 are the number of samples in class 1 and 2. The expression of the j-th gene is modeled as $N(\hat{\mu}_{j1}, \hat{\sigma}_{j1})$ or $N(\hat{\mu}_{j2}, \hat{\sigma}_{j2})$ depending on whether the sample belongs to class 1 or 2. We define the class membership likelihood of expression x_j^i from the models of classes 1 and 2 as p_{j1} and p_{j2} respectively.

$$p_{j1} = \mathbb{P}(\xi \geq x_j^i \,|\, \xi \sim \mathbf{N}(\hat{\mu}_{j1}, \hat{\sigma}_{j1})), \quad \text{if } x_j^i \geq \hat{\mu}_{j1},$$
$$p_{j1} = \mathbb{P}(\xi < x_j^i \,|\, \xi \sim \mathbf{N}(\hat{\mu}_{j1}, \hat{\sigma}_{j1})), \quad \text{if } x_j^i < \hat{\mu}_{j1},$$
$$p_{j2} = \mathbb{P}(\xi \geq x_j^i \,|\, \xi \sim \mathbf{N}(\hat{\mu}_{j2}, \hat{\sigma}_{j2})), \quad \text{if } x_j^i \geq \hat{\mu}_{j2},$$
$$p_{j2} = \mathbb{P}(\xi < x_j^i \,|\, \xi \sim \mathbf{N}(\hat{\mu}_{j2}, \hat{\sigma}_{j2})), \quad \text{if } x_j^i < \hat{\mu}_{j2}.$$

We use the distribution function rather than the density because there is a very natural directionality assumption in this model in that if the Gaussians are well separated then the deeper inside the respective class a point x resides the higher should be the membership probability. We then use the log-likelihood ratio as the correlation statistic. Given expression, x_j^i, of the j-th gene of the i-th sample the correlation statistic is computed as:

$$c_j^i = \log\left(\frac{p_{j1}}{p_{j2}}\right), \quad \text{if } \hat{\mu}_{j1} \geq \hat{\mu}_{j2}$$
$$c_j^i = \log\left(\frac{p_{j2}}{p_{j1}}\right), \quad \text{otherwise.}$$

Thus, genes are ranked based upon the differential probability of their membership in either class and because of this, genes are ranked as a continuum from those with the greatest probability of belonging to class 1 ranked at the top and genes with the greatest probability of belonging to class 2 near the bottom. As most genes will have limited differential expression between the two classes, these genes will have similar probabilities of belonging to either group and the log-likelihood ratio will be near zero.

2.1.2 Nonparametric model The assumption of normality in the parametric model is often inappropriate for expression data. For this reason, a nonparametric model to compute class membership likelihoods is used in most applications. The class membership likelihoods are computed based upon absorption probabilities of a Brownian motion (random walk) (see Figure 1 for an illustration of the model).

We first estimate the densities of the j-th gene for the two classes, $\hat{p}_{j1}(x)$ and $\hat{p}_{j2}(x)$, using a Parzen estimator (Vapnik, 1998) with a Gaussian kernel:

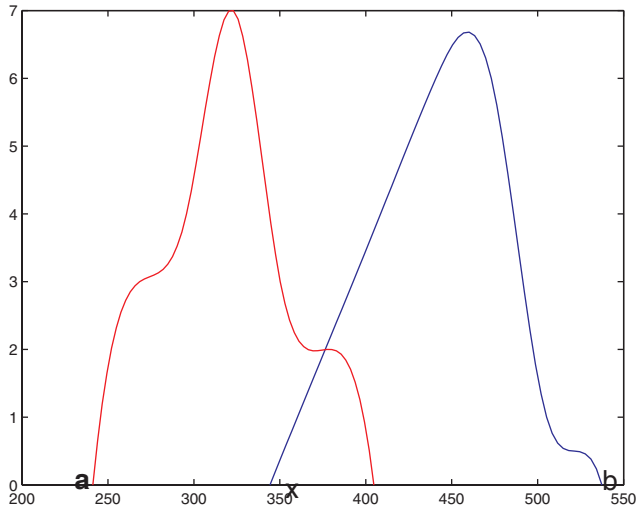


Fig. 1. The two classes' densities are displayed by the red and blue curves, p_r, p_b. Assume we have a diffusion process (random walk) starting at x. We compute the probability of absorption at the point b if the initial conditions are distributed as p_b, $\mathbb{P}$ (absorption at b starting at x $|\, p_b$). We also compute the probability of absorption at the point a if the initial conditions are distributed as p_r, $\mathbb{P}$ (absorption at a starting at x $|\, p_r$). These two probabilities serve as a measure that an individual sample belongs to one of the two distributions.

$$\hat{p}_{j1}(x) = \frac{1}{n_1\sigma_{j1}\sqrt{2\pi}}\sum_{i\in C_1} e^{-|x_j^i - x|^2/2\sigma_{j1}^2},$$
$$\hat{p}_{j2}(x) = \frac{1}{n_2\sigma_{j2}\sqrt{2\pi}}\sum_{i\in C_2} e^{-|x_j^i - x|^2/2\sigma_{j2}^2},$$

where n_1 and n_2 are the number of samples in C_1 and C_2 and bandwidths σ_{j1} and σ_{j2} are set to the average distance between points of the j-th gene in C_1 and C_2 respectively. We define two points, e_{j1}^i and e_{j2}^i, as the left or right extremes of the random walk starting at x_j^i.

$$e_{j1}^i = \min_i\{x_j^i\} \text{ if } x_j^i < \hat{\mu}_{j1}, \quad e_{j1}^i = \max_i\{x_j^i\} \text{ if } x_j^i \geq \hat{\mu}_{j1}$$
$$e_{j2}^i = \min_i\{x_j^i\} \text{ if } x_j^i < \hat{\mu}_{j2}, \quad e_{j2}^i = \max_i\{x_j^i\} \text{ if } x_j^i \geq \hat{\mu}_{j2}.$$

The membership likelihood of expression x_j^i for class 1and 2 is given by the absorption probability at the points $e_{j1}^i$and e_{j2}^i for a Brownian motion starting at x_j^i with initial conditions distributed as $\hat{p}_{j1}(x)$ and $\hat{p}_{j2}(x)$.

We again use the log-likelihood ratio as the correlation statistic. Given expression, x_j^i, of the j-th gene of the i-th sample the correlation statistic is computed as:

$$c_j^i = \log\left(\frac{\mathbb{P}(\text{absorption at } e_{j1}^i \text{ starting at } x_j^i \,|\, \hat{p}_{j1})}{\mathbb{P}(\text{absorption at } e_{j2}^i \text{ starting at } x_j^i \,|\, \hat{p}_{j2})}\right). \qquad (3)$$

The absorption probabilities can be computed as the solution of the Dirichlet problem (Durrett, 1996) which for the Parzen estimators results in a weighted sum of error functions and exponentials (see methods section for the exact form and derivation). So the correlation statistics can be computed efficiently.

2.2. Validation on Model Systems

The objective of ASSESS is to annotate each sample in an expression data set in terms of a priori defined gene sets often constructed

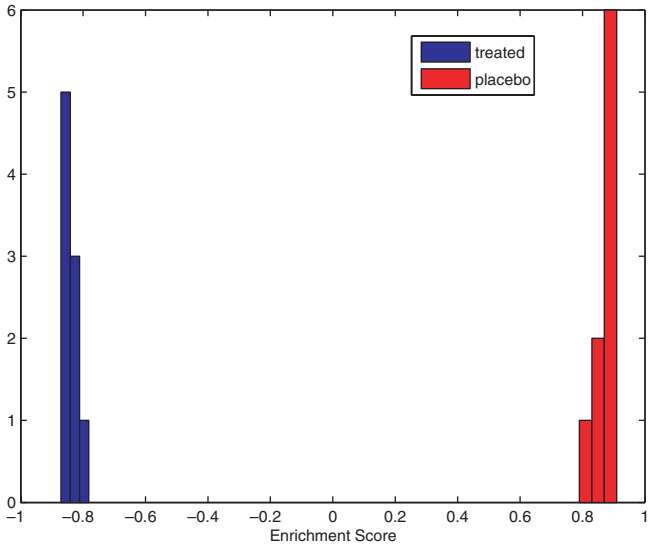


Fig. 2. A histogram of enrichment scores for the 9 treated and 9 untreated mouse prostate samples in the test data with respect to the AKT pathway gene set computed from the training data.

from model systems. In this section we validate the method by demonstrating that gene sets built from model systems or with known genetic perturbations are indeed enriched in gene expression data from the same model systems or related systems.

2.2.1 Mouse models In (Majumder *et al*., 2004) transgenic mice were generated that developed a highly penetrant prostatic intraepithelial neoplasia (PIN) phenotype and expressed a constitutively active AKT1 gene in the ventral prostate of the mouse. This AKT-induced PIN phenotype can be reversed with treatment of RAD001, a mTOR inhibitor. The transgenic mice were split into two groups, with one group receiving RAD001 and the other a placebo. Tissue was taken from the prostate of both groups and DNA microarray analysis was preformed using the Affymetrix Murine U430A microarray. This resulted in two sets of expression data: samples treated with RAD001 ($n = 19$) and placebo ($n = 19$). These data sets were split into a training and test set. The training set consisted of the first 10 samples treated with RAD001 and the first 10 samples treated with the placebo. The test set was comprised of the complimentary samples. The training set was used to construct an AKT gene set using a logistic regression model (see methods section for details).

We applied ASSESS to the test set using the AKT gene set derived from the training data. The enrichment scores of the samples treated with RAD001 strongly indicate that genes in the AKT gene set were under expressed compared to the samples given placebo which showed enrichment in the gene set (see Figure 2). All samples were significantly enriched (p-value < 0.001).

2.2.2 Cell culture models In (Bild *et al*., 2006) human primary mammary epithelial cell cultures (HMECs) were used to develop a series of pathway signatures which were then used to assay pathway disregulation in non-small cell lung carcinoma (NSCLC). We use this data set to validate our method.

Table 1. Average enrichment scores (±sd) for the comparison of normal (GFP cells to infected cells for the gene set built form the respective infected cell type

Experiment	ES for GFP cells	ES for infected cells
BCAT	−0.87(±0.031)	0.88(±0.042)
E2F3	−0.97(±0.0069)	−0.98(±0.0061)
MYC	−0.89(±0.018)	−0.91(±0.067)
RAS	−0.96(±0.012)	−0.91(±0.021)
SRC	−0.90(±0.016)	−0.91(±0.022)

The data was generated by using recombinant adenoviruses to express specific oncogenes in an otherwise quiescent cell, thereby isolating the subsequent events as defined by the activation/deregulation of that single pathway. The cells were infected with adenovirus expressing either human c-Myc, activated H-Ras, human c-Src, human E2F3, or activated β-cantenin. RNA from these multiple independent infections, as well as from normal cells (with green florescent protein, GFP), was collected for DNA microarray analysis using the Affymetrix Human Genome U133 Plus 2.0 Array.

Given the independent replicates from the six conditions, the five perturbed pathways and the normal GFP cells, we split each condition into a train and test set. Thus given expression data from: 10 Myc, 10 Ras, 7 Src, 10 E2F3, 9 β-catenin, and 10 normal/GFP samples we construct five training sets with the first half of the samples from each experimental data set along with the first 5 normal samples. Similarly, five independent test sets were constructed using the complimentary samples (the second half of samples in the six conditions). The training sets were used to construct gene sets for each of the five pathways, Myc, Ras, Src, E2F3, and β-catenin using a logistic regression model (see methods section for details).

ASSESS was applied to the five test sets to calculate enrichment with respect to the five gene sets computed from the training data.

2.2.3 Literature based models The approach developed in (Huang *et al*., 2003; Black *et al*., 2003; Bild *et al*., 2006) of building statistical models of pathway deregulation in controlled experiments and then applying these to new data sets could have been used in the previous two examples. However, this approach requires that the cell line perturbation data as well as new data and that the data are on comparable platforms. This approach cannot be used for gene sets derived from literature whereas ASSESS is still applicable.

In (Subramanian *et al*., 2005) a data set generated from mRNA expression from lymphoblastoid cell lines derived from 15 males and 17 females served as a validation set. We then sought to address the following: which gene sets were over expressed in males and which in females. Gene sets defined by cytogenetic bands and gene sets defined by pathway or functional properties were examined. As expected for males, chromosome Y as well as its two bands (Yp11 and Yq11) and a gene set corresponding to genes enriched in male reproductive tissue (testis) were overexpressed. For females two gene sets of genes that escape X-inactivation were overexpressed in addition to a gene set corresponding to genes enriched in female reproductive tissue (uterus). Genes on the X-chromosome would

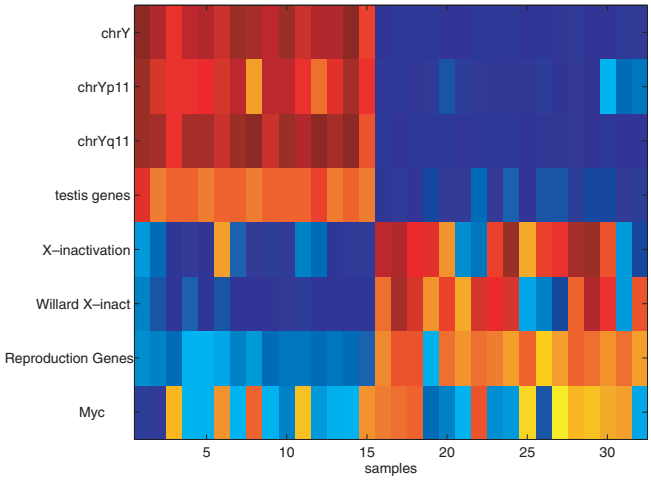


Fig. 3. Enrichment scores for the comparison of males to females in the 8 gene sets. The male samples ($1-15$) show enrichment in the Y, Yq11, Yp11, and testis gene sets. The female samples ($16-32$) show enrichment in the two escape of X-inactivation gene sets and the uterus gene set. The Myc pathway shows no differential expression between males and females, as expected.

not be expected to be overexpressed due to dosage compensation by X-inactivation.

The enrichment of these seven gene sets with respect to the male and female samples in the lymphoblastoid cell lines is displayed in Figure 3. The male samples are clearly enriched with respect to: Y, Yq11, Yp11, and testis. The female samples are clearly enriched with respect to: the two escape of X-inactivation gene sets (X-inactivation and Willard X-inactivation) and the uterus gene set (labeled in the figure as Reproduction Genes). We used a Myc gene set as a control in that it is not expected to be enriched with respect to the male/female distinction and indeed this is the case.

We further illustrate the procedure by plotting the random walk (Equation (1)) for a male and female sample with respect to one of the escape from X-inactivation gene sets and a Myc gene set (see Figure 4). For a female sample with respect to this gene set, the random walk increases very rapidly initially indicating that genes escaping X-inactivation appear at the top of the list of genes ordered by correlation with the female phenotype. This results in a very positive enrichment score. For the male sample the random walk is basically a mirror image of the female case indicating that genes escaping X-inactivation appear at the end of a list of genes ordered by correlation with the male phenotype. This results in a very negative enrichment score. The third case is for a female sample with respect to the Myc gene set. In this case the genes in the gene set are randomly spread over the ranked list and so the enrichment score never deviates far from zero.

2.3 Classification and clustering in the space of pathways

A very natural consequence of obtaining enrichment scores for each sample in the data set is that classification and clustering can now be performed in the space of gene sets rather than individual genes. Being able to interpret classification models using

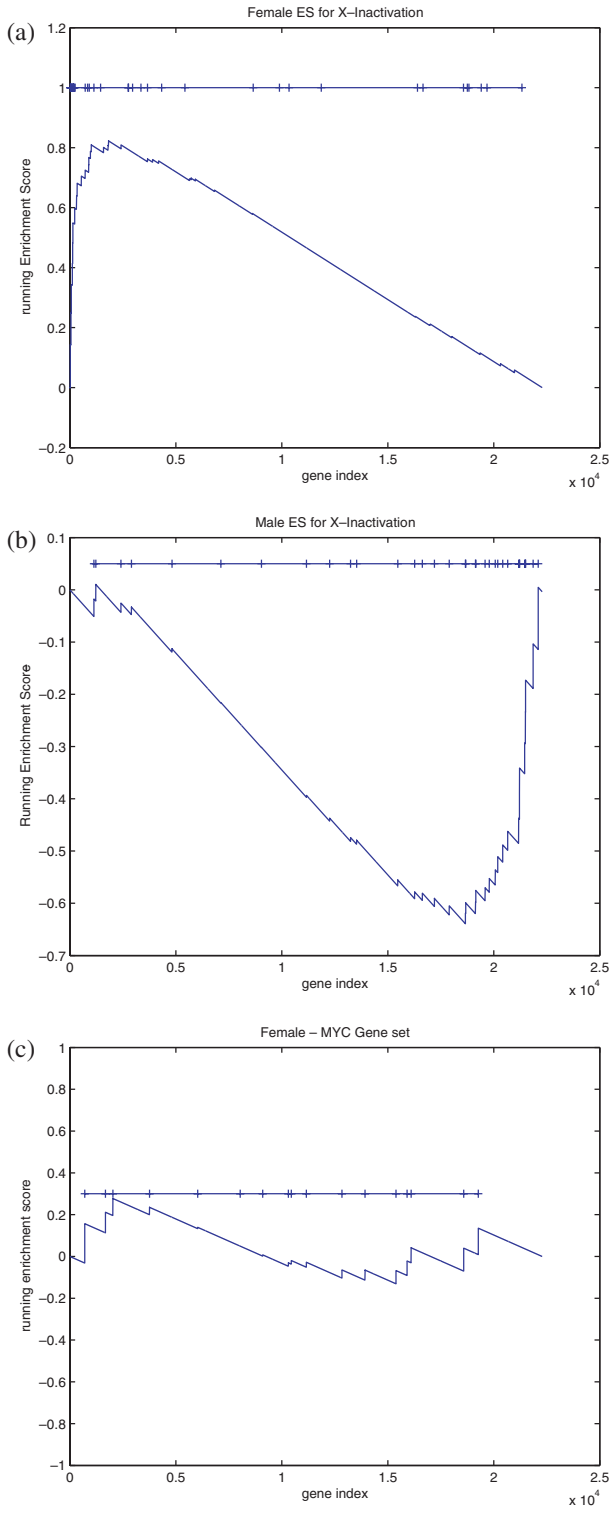


Fig. 4. Random walks. (a) The random walk for a female sample with respect to one of the escape from X-inactivation gene sets. The hatches of the top line indicate where the genes in the gene set fall with respect to the rank-ordering. (b) The random walk for a male sample with respect to one of the escape from X-inactivation gene sets. (c) The random walk for female sample with respect to a Myc gene set. This gene set is not significantly enriched and so the hatches appear randomly dispersed with respect to the rank-ordering.

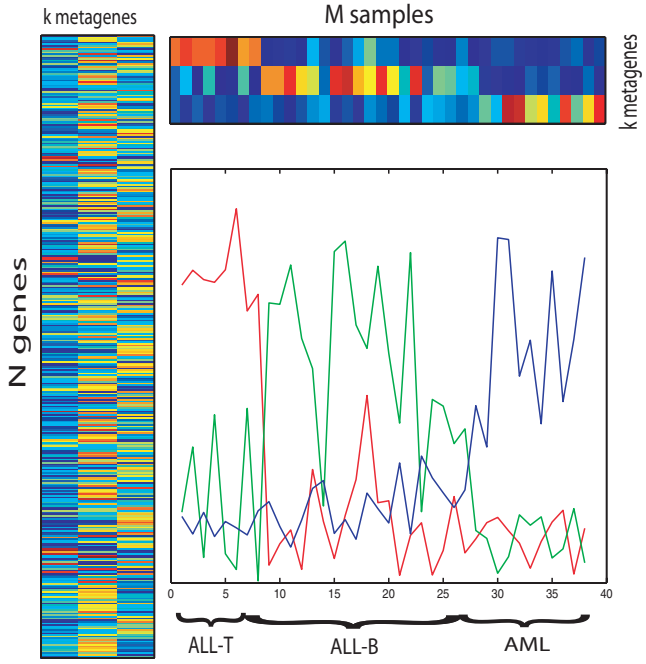


Fig. 5. The top and left figure are the left and right matrix factors for the matrix of enrichment scores in the Leukemia data with $k = 3$. The red line is a plot of the first metapathway over the data and this metapathway selects for the ALL-T samples. The green line is the second metapathway and it selects for the ALL-B samples. The blue line is a plot of the third metapathway which selects for the AML samples.

pathways offers an alternative and possibly more intuitive perspective than models using individual genes. Another aspect of building models in the space of pathways that was emphasized in (Bild *et al*., 2006) is knowing which pathways are disregulated with respect to outcome and how this can help suggest targeted therapeutics.

2.3.1 Clustering In (Brunet *et al*., 2004) a matrix factorization method (NMF) was applied to an expression data set with acute myelogenous leukemia (AML) and acute lymphoblastic leukemia (ALL) samples (Golub *et al*., 1999). The matrix factorization allowed the clustering of the samples into subsets. A parameter in this clustering method is the number of subsets k. For this data set results with $k = 2, 3$ were computed. For the two cases the clusters comprised of $\{(25\ \text{ALL}),\ (11\ \text{AML}, 2\ \text{ALL})\}$ and $\{(8\ \text{ALL-T}), (17\ \text{ALL-B}), (11\ \text{AML}, 2\ \text{ALL-B})\}$, where ALL-T and ALL-B are two subtypes of ALL. We applied ASSESS to this leukemia data set using a database of 523 gene sets (Subramanian *et al*. 2005). We then applied NMF to this space of enrichment scores and obtained identical results. The only difference is according to the measure of confidence developed in (Brunet *et al*., 2004), as the clustering obtained from the enrichment space had greater confidence than that from the raw expression data. The result of the clustering and the factors are displayed in Figure 5.

2.3.2 Classification We examined six gene expression data sets for which single gene classification models have been built: (a) Gender – male vs. female (Subramanian *et al*., 2005), (b) cDNA Lung cancer – squamous vs. adenocarcinoma (Garber *et al*., 2001), (c) oligonucleotide Lung cancer – squamous vs. adenocarcinoma (Potti *et al*., 2006), (d) Medulloblastoma – survival vs. failure (Pomeroy *et al*., 2002), (e) Prostate cancer – recurrence vs. nonrecurrence (Glinsky *et al*., 2004), and (f) Leukemia – AML vs. ALL (Golub *et al*., 1999).

Table 2. Classification accuracy for six data sets building classification models in the space of enrichment scores

Classes	Accuracy
Gender: males vs. Females	94%
Lung Cancer(cDNA): Adeno vs. Squamous	91%
Lung Cancer(oglio): Adeno vs. Squamous	84%
Medulloblastoma: survival vs. failure	72%
Prostate: recurrent vs. nonrecurrent	73%
Leukemia: ALL vs. AML	85%

We applied the classification using enrichment scores procedure outlined in the methods section to compute classification accuracy on these six data sets (see Table 2). For all the data sets except for the Leukemia data set the leave-one-out method was used (Algorithm 1). For the Leukemia data set the train-test procedure was used (Algorithm 2) with the train-test split outlined in (Golub *et al*., 1999). The classification accuracy was comparable or better than that for single gene classifiers for all the data sets except for the Leukemia data.

The pathways associated with recurrent prostate cancer tumors supports ASSESS's ability to both accurately predict outcome as well as provide biological insight. Both AKT and PTEN gene sets were found to have increased coordinate expression in samples of recurrent prostate cancer. PTEN loss is one of the most common genetic alterations seen in advanced prostate cancer resulting in activation of the PI3K-AKT pathway. Activation of this pathway is known to occur at a greater frequency in advanced prostate cancer and has prognostic significance. A ''TERT-up'' gene set was similarly found to be associated with recurrent prostate cancer. An essential requirement for tumor progression is avoidance of cellular senescence, telomerase restores chromosomal telomeres and is associated with the development of prostate cancer. Finally, another interesting observation is the presence of the ''DNA damage signaling'' and ''Cell cycle checkpoint'' pathways both representing common cellular processes disregulated in aggressive cancer.

2.4 Cross-platform expression models

DNA microarray studies have been carried out on a variety of platforms for the same case-control experiment, for example both cDNA microarrays and oligonucleotide microarrays are popular in cancer genomics. The integration of data across platforms is appealing for a variety of reasons: increasing the sample size of the data, allowing for interstudy validation, mitigating platform based biases, and mitigating study based biases.

Building a model from raw expression data from one platform and applying the model to data from another platform directly will not work since the expression data from the two platforms have different distributions. One approach to normalize between the platforms is to use median rank scores and quantile discretization to map the data to a common space and then build a classification model in this space (Warnat *et al*., 2005).

We advocate an alternative approach of applying ASSESS to expression data to map the data into the space of enrichment scores for pathways and then building models in this space. Methods such as median rank scores are no longer needed as enrichment scores between platforms are numerically comparable. There are several advantages to this approach: (1) the need to map genes using UniGene ids is avoided; (2) the problem of multiple probe mappings between platforms is avoided; (3) gene sets defined separately by probes specific to each platform can be used; (4) the enrichment statistic is much more robust than the rank of a single gene so the loss of genes between platforms is mitigated; (5) interpreting results on the level of pathways instead of single genes is appealing.

We first applied this approach to two prostate cancer studies (Welsh *et al*., 2001; Dhanasekaran *et al*., 2001). The two platforms for the studies were cDNA microarrays (Dhanasekaran *et al*., 2001) and Affymetrix oligonucleotide microarrays (Welsh *et al*., 2001). The cDNA data set contained 53 samples of which 34 were tumors and 19 were normal. The oligo data set contained 33 samples of which 24 were tumors and 9 were normal. The catalog of human functional gene sets comprised of 433 sets annotated for both platforms (Subramanian *et al*., 2005) was used as the gene set. The error rate for using the cDNA and oligo data sets as train-test sets respectively is reported in Table 3, as is the error rate for a leave-one-out procedure using all the cDNA and oligo samples (see methods section for details on both test-train and leave-one-out classification using gene sets). We compare these results with the leave-one-out error computed on the individual data sets (see Table 3).

We next applied this approach to two lung cancer studies (Garber *et al*., 2001; Potti *et al*., 2006). The two studies involved the same platforms as the prostate example. The cDNA data set contained 55 samples of which 38 were adenocarcinomas and 17 were squamous cell lung carcinomas (Garber *et al*., 2001). The oligo data set contained 93 samples of which 45 were adenocarcinomas and 48 were squamous cell lung carcinomas (Potti *et al*., 2006). The same catalog of gene sets as used in the prostate example was used. The error rates for the cross-platform predictions as well as predictions within the individual data sets are summarized in Table 3.

Table 3. Classification accuracy for cross-platform models for the prostate and lung cancer data sets

	cDNA LOO	oligo LOO	train-test	combined LOO
prostate T/N	85.7%	76.5%	(cDNA-oligo) 73.5%	80.7%
lung A/S	88.0%	90.9%	(oligo-cDNA) 78.2%	88.5%

Algorithm 1: Leave-one-out procedure for pathway based classification.

input: training data and gene sets
return: error rate
for $i = 1$ **to** n **do**
  split the data into x_i (the i-th data point) and $X^{\setminus i}$ (the data with the i-th point removed);
  compute Tr $= ES_i^k$ for $X^{\setminus i}$ (this is the enrichment of the m gene sets on the $n-1$ data in $X^{\setminus i}$);
  compute Test=ES_i^k for x_i (this is the enrichment of the m gene sets on x_i, the label i-th point is not used in the estimation of the enrichment score by leaving this point out of the Parzen estimator);
  use Tr to build logistic regression with variable selection M_i;
  apply M_i to Test to obtain prediction $\hat{y}_i$;
  if $y :\neq \hat{y}_i$ **then** error rate = error rate + 1;
return error rate

Algorithm 2: Test error extimate for pathway based classification.

input: training data, test data, and gene sets
return: error rate
compute Tr $= ES_i^k$ for X (this is the enrichment of the m gene sets on the training data X);
use Tr to build logistic regression with variable selection model M;
for $j = 1$ **to** n' **do**
  compute Test $= ES_j^k$ for z_j (this is the enrichment of the m gene sets on the j-th test sample, use only the training data X to compute the Parzen estimator);
  apply M to Test to obtain prediction $\hat{t}_j$;
  if $t_j \neq \hat{t}_j$ **then** error rate = error rate + 1;
return error rate

3 METHODS

3.1 Gene set construction

Given an expression data set with two class labels, we use a linear logistic regression model with regularization or shrinkage (Hastie *et al*., 2000) to construct gene sets. We define the expression data as a matrix x_j^i with $i = 1, \ldots, n$ (the number of samples) and $j = 1, \ldots, p$ (the number of genes), the i-th sample is designated as x_i, and the class labels as $y \in \{-1, 1\}$. The logistic regression model with regularization involves solving the following optimization problem

$$\arg\min_{w,b}\left[\frac{1}{n}\sum_{i=1}^{n}\log(1+e^{-(y_i(w\cdot x_i+b))})+\lambda\|w\|^2\right],$$

where λ is a model parameter that needs to be set.

Solving the above optimization problem results in a vector $\hat{w}$ and the absolute magnitude of the elements of the vector correspond to the relevance of a gene or feature. For the HMEC data sets and the AKT data set, genes corresponding to 50 elements of $\hat{w}$ most correlated with the perturbation phenotype were used to construct the gene sets. In both algorithms 1 and 2 genes corresponding to the top and bottom 50 elements of $\hat{w}$ were used.

3.2 Classification and gene set selection

Classification using enrichment scores was applied in two settings: a train-test setting and a leave-one-out cross-validation setting. The leave-one-out setting was used for all the data sets except the leukemia data set for which we used the test-train setting. The test-train setting is a simple generalization of the leave-one-out setting.

3.2.1 Leave-one-out setting Given data set x_j^i of gene expression for $j = 1, \ldots, p$ genes and $i = 1, \ldots, n$ samples where the i-th column of the matrix X correspond to the i-th sample, labels $(y_i)_{i=1}^n$, and gene sets $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ the leave-one-out method outlined in Algorithm 1 provides an unbiased estimate of the error rate (technically leave-one-out estimators are almost unbiased (Vapnik, 1998)).

Table 4. Probability of class membership as a function of x_j^x and the class means

$\mathbb{P}(\hat{x}_j^i \in C_1)$	$\hat{x}_j^i \geq \hat{\mu}_{j1}$	$\mathbb{P}$(absorption at 0 starting at $\hat{x}_j^i \mid \hat{p}_{j1}$
	$\hat{x}_j^i > \hat{\mu}_{j1}$	$\mathbb{P}$(absorption at 1 starting at $\hat{x}_j^i \mid \hat{p}_{j1}$
$\mathbb{P}(\hat{x}_j^i \in C_2)$	$\hat{x}_j^i \leq \hat{\mu}_{j1}$	$\mathbb{P}$(absorption at 0 starting at $\hat{x}_j^i \mid \hat{p}_{j2}$
	$\hat{x}_j^i < \hat{\mu}_{j1}$	$\mathbb{P}$(absorption at 1 starting at $\hat{x}_j^i \mid \hat{p}_{j2}$

3.2.2 Train-test setting Given a training set of $X = (x_i)_{i=1}^n$ expression profiles and labels $(y_i)_{i=1}^n$, a test set of $Z = (x_j)_{j=1}^{n'}$ expression profiles with labels $(t_j)_{j=1}^{n'}$, and gene sets $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ the procedure outlined in Algorithm 2 provides an unbiased error estimate on the test set.

3.3 Computation of absorption probabilities

To compute the correlation coefficients in the nonparametric model we need to compute the probability that the expression of the j-th gene in the i-th sample is representative of class 1 or class 2, $\mathbb{P}(x_j^i \in C_1)$ and $\mathbb{P}(x_j^i \in C_2)$ for all samples $i = 1, \ldots, n$ and genes $j = 1, \ldots, p$. We first scale the expression data for each gene to $[0, 1]$,

$$\hat{x}_j^i = \frac{x_j^i - \min_i(x_j^i)}{\max_i(x_j^i) - \min_i(x_j^i)}.$$

The class membership probabilities are the probabilities of absorption to the left or right extreme, which are $\{0, 1\}$ for the scaled data, depending on whether $\hat{x}_i^j$ is greater or less than the scaled class means (see Table 4). This simply reflects the directionality assumption of our model.

Let

$$u(\hat{x}) = \mathbb{P}(\text{absorption at 0 starting at } \hat{x}),$$
$$v(\hat{x}) = \mathbb{P}(\text{absorption at 1 starting at } \hat{x}),$$

and let $p(\hat{x})$ be supported on $[0, 1]$, then

$$\mathbb{P}(\text{absorption at 0 starting at } \hat{x} \mid p(\hat{x})) = \int_0^{\hat{x}} u(\hat{x})\, p(\hat{x})\, d\hat{x},$$
$$\mathbb{P}(\text{absorption at 1 starting at } \hat{x} \mid p(\hat{x})) = \int_{\hat{x}}^1 v(\hat{x})\, p(\hat{x})\, d\hat{x}.$$

The absorption probabilities of a Brownian motion at the end points of a line segment can be computed by solving the heat equation with appropriate boundary conditions, the Dirichlet problem (Durrett, 1996). In the above case we solve for

$$\frac{d^2u(\hat{x})}{d\hat{x}^2} = 0 \quad s.t. \quad u(0) = 0,\ u(1) = 1$$
$$\frac{d^2v(\hat{x})}{d\hat{x}^2} = 0 \quad s.t. \quad av(0) = 1,\ v(1) = 0.$$

This results in the solutions

$$u(\hat{x}) = \hat{x}, \qquad v(\hat{x}) = 1 - \hat{x}.$$

Given the Parzen estimates of the densities for the two classes

$$\hat{p}_{j1}(\hat{x}) = \frac{1}{n_1\sigma_{j1}\sqrt{2\pi}} \sum_{i \in C_1} e^{-|\hat{x}_j^i - \hat{x}|^2/2\sigma_{j1}^2},$$
$$\hat{p}_{j2}(\hat{x}) = \frac{1}{n_2\sigma_{j2}\sqrt{2\pi}} \sum_{i \in C_2} e^{-|\hat{x}_j^i - \hat{x}|^2/2\sigma_{j2}^2},$$

we can compute the absorption probabilities as

$$\mathbb{P}(\text{absorption at 0 starting at } \hat{x} \mid \hat{p}_{jc}) = \int_0^{\hat{x}} s\, \hat{p}_{jc}(s)\, ds$$
$$\mathbb{P}(\text{absorption at 1 starting at } \hat{x} \mid \hat{p}_{jc}) = \int_{\hat{x}}^1 (1 - s)\, \hat{p}_{jc}(s)\, ds$$

where c denotes the classes $\{1, 2\}$. Solving the integrals results in a weighted sum of error functions and exponentials.

4 DISCUSSION

In this paper we introduce a formal statistical method to measure the enrichment of each sample in an expression data set with respect to a priori defined gene sets. This allows us to assay the natural variation of pathway activity in observed gene expression data sets. It is a natural extension of methods that measure the enrichment of an entire data set with respect to a priori defined gene sets (Subramanian *et al*., 2005; Barry *et al*., 2005; Kim and Volsky, 2005; Tomfohr *et al*., 2005).

The method was validated on a variety of model systems: oncogenic cell lines, mouse models, and known gender differences in expression. The utility of the method was demonstrated by clustering and building classification models in the space of pathways or gene sets. These were in general as accurate as methods applied in the space of genes but more interpretable and robust. This robustness was illustrated by the ability to build models between different expression based technologies— cross-platform models. This is hard to do in the single gene setting.

A variety of open questions regarding the pathway paradigm and our implementation of it remain. Some of these questions are technical and some are fundamental with respect to both statistical analysis and molecular biology.

We first discuss the technical issues:

- Enrichment statistic: We use a maximum deviation statistic to compute our enrichment score. The theory behind BLAST (Ewans and Grant, 2002) offers insights as to how we may improve our statistic by adding to the maximal extremal excursion the top r excursions. This would especially make sense when the gene set corresponds to genes in a pathway that subdivide into sub-pathways, some of which are up regulated and some of which are down regulated.
- Correlation statistic: We used a Brownian motion model to compute our correlation statistic. This outperformed a simple Gaussian model and a model based upon the cumulative distribution function of the Parzen estimator. However, these models are by no means exhaustive and other statistics may be as robust but with greater sensitivity.
- Extension to real-valued phenotypes: We stated the procedure for the case with binary phenotypes. The crux of an extension to real-valued phenotypes would be the computation of an appropriate correlation statistic. In the context of a survival model this would not be difficult but in general it can be complicated.

There are two fundamental questions with respect to our approach and they are intimately related

- What is a pathway (gene set): Gene sets can be derived from experimental perturbations, literature based studies, and a variety of other origins. A fundamental question is which of these sets is most appropriate. For example, a database of gene sets may contain 5 Ras pathways from a variety of experiments or literature surveys. For a particular analysis which is most appropriate? The authors believe that a partial answer or consensus is developing that experimentally based gene sets are in general more robust than ones derived from literature. However, the quantification of this and a statistically formal method for scoring gene sets is still an open problem.

- Likelihood based testing: The statistic used in our hypothesis testing framework is likelihood based, $\mathbb{P}(\text{data} \mid \text{pathway})$. The problem with using this likelihood based framework is that in this formulation is that the pathway we condition upon is not fixed. The Ras pathway as defined today is different than the Ras pathway as defined in two weeks, some genes are added and some removed. In the above framework one can then ask which pathway are we testing, is there multiplicity in the Ras pathway and if so how many Ras pathways are there. An alternative approach which is conceptually very appealing is to build our statistical framework on the posterior, $\mathbb{P}(\text{pathway} \mid \text{data})$. This provides a uniform framework and quantity that we can use to score the different Ras pathways in the previous example. The fundamental problem in using the posterior is that a prior is needed on the space of pathways (for example priors over possible Ras pathways). The construction or estimation by sampling gene expression data sets of such a priori defined gene sets starting with a database of pathways is a very interesting and challenging computational biology and statistics problem.

ACKNOWLEDGEMENT

E.E. receives support from a National Institutions of Health training grant. P.G.F. is a Damon Runyon Cancer Research Foundation Clinical Investigator and receives support from the Prostate Cancer Foundation. P.G.F. and S.M. both acknowledge the Institute for Genome Sciences & Policy for support. S.M. would like to thank Ran Liu for her help early on in this project. S.M. would like to thank Jonathan Mattingly for discussions.

REFERENCES

Alvarez,J., Febbo,P., Ramaswamy,S., Loda,M., Richardson,A. and Frank,D. (2005) Identification of a genetic signature of activated signal transducer and activator of transcription 3 in human tumors. *Cancer Res*, **65**, 5054–62.

Barry,W., Nobel,A. and Wright,F. (2005) Significance analysis of functional categories in gene expression studies: a structured permutation approach. *Bioinformatics*, **21**, 1943–9.

Bild,A., Yao,G., Chang,J., Wang,Q., Potti,A., Chasse,D., Joshi,M., Harpole,D., Lancaster,J., Berchuck,A., Olson,J., Marks,J., Dressman,H., West,M. and Nevins,J. (2006) Oncogenic pathway signatures in human cancers as a guide to targeted therapies. *Nature*, **439**, 353–357.

Black,E., Huang,E., Dressman,H., Rempel,R., Laakso,N., Asa,S., Ishida,S., West,M. and Nevins,J. (2003) Distinct gene expression phenotypes of cells lacking Rb and Rb family members. *Cancer Res*, **63**, 3716–23.

Brunet,J., Tamayo,P., Golub,T. and Mesirov,J. (2004) Metagenes and molecular pattern discovery using matrix factorization. *Proc Natl Acad Sci USA*, **101**, 4164–9.

Dhanasekaran,S., Barrett,T., Ghosh,D., Shah,R., Varambally,S., Kurachi,K., Pienta,K., Rubin,M. and Chinnaiyan,A. (2001) Delineation of prognostic biomakers in prostate cancer. *Nature*, **412**, 822–826.

Durrett,R. (1996) *Stochastic Calculus: A Practical Introduction*. CRC Press, Boca Raton, FL.

Ewans,W. and Grant,G. (2002) *Statistical Methods in Bioinformatics*. Springer.

Febbo,P., Richie,J., George,D., Loda,M., Manola,J., Shankar,S., Barnes,A., Tempany,C., Catalona,W., Kantoff,P. and Oh,W. (2005) Neoadjuvant docetaxel before radical prostatectomy in patients with high-risk localized prostate cancer. *Clin Cancer Res*, **11**, 5233–40.

Feller,W. (1971) *An Introduction to Probability Theory and Its Applications, Vol 1*. John Wiley & Sons, New York.

Garber,M., Troyanskaya,O., Schluens,K., Petersen,S., Thaesler,Z., Pacyna-Gengelbach,M., van de Rijn,M., Rosen,G., Perou,C., Whyte,R., Altman,R., Brown,P., Botstein,D. and Petersen,I. (2001) Diversity of gene expression in adenocarcinoma of the lung. *Proc Natl Acad Sci USA*, **98**, 13784–9.

Glinsky,G., Glinskii,A., Stephenson,A., Hoffman,R. and Gerald,W. (2004) Gene expression profiling predicts clinical outcome of prostate cancer. *J Clin Invest*, **113**, 913–23.

Golub,T., Slonim,D., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J., Coller,H., Loh,M., Downing,J., Caligiuri,M., Bloomfield,C. and Lander,E. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–7.

Hastie,T., Tibshirani,R. and Friedman,J. (2000) *The Elements of Statistical Learning, Data Mining, Inference and Prediction*. Springer Verlag, New York.

Huang,E., Ishida,S., Pittman,J., Dressman,H., Bild,A., Kloos,M., D'Amico,M., Pestell,R., West,M. and Nevins,J. (2003) Gene expression phenotypic models that predict the activity of oncogenic pathways. *Nat Genet*, **34**, 226–30.

Kim,S. and Volsky,D. (2005) PAGE: Parametric Analysis of Gene Set Enrichment. *BMC Bioinformatics*, **6**.

Majumder,P., Febbo,P., Bikoff,R., Berger,R., Xue,Q., McMahon,L., Manola,J., Brugarolas,J., McDonnell,T., Golub,T., Loda,M., Lane,H. and Sellers,W. (2004) mTOR inhibition reverses Akt-dependent prostate intraepithelial neoplasia through regulation of apoptotic and HIF-1-dependent pathways. *Nat Med*, **10**, 594–601.

Mootha,V., Lindgren,C., Eriksson,K., Subramanian,A., Sihag,S., Lehar,J., Puigserver,P., Carlsson,E., Ridderstrale,M., Laurila,E., Houstis,N., Daly,M., Patterson,N., Mesirov,J., Golub,T., Tamayo,P., Spiegelman,B., Lander,E., Hirschhorn,J., Altshuler,D. and Groop,L. (2003) Pgc-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat Genet*, **34**, 267–73.

Pomeroy,S., Tamayo,P., Gaasenbeek,M., Sturla,L., Angelo,M., McLaughlin,M., Kim,J., Goumnerova,L., Black,P., Lau,C., Allen,J., Zagzag,D., Olson,J., Curran,T., Wetmore,C., Biegel,J., Poggio,T., Mukherjee,S., Rifkin,R., Califano,A., Stolovitzky,G., Louis,D., Mesirov,J., Lander,E. and Golub,T. (2002) Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, **415**, 436–42.

Potti,A., Mukherjee,S., Petersen,R., Dressman,H., Bild,A., Koontz,J., Kratzke,R., Watson,M., Kelley,M., Ginsburg,G., West,M., Harople,D.J. and Nevins,J. (2006) A Genomic Strategy to Refine Prognosis in Early Stage Non-Small Cell Lung Carcinoma. Submitted.

Subramanian,A., Tamayo,P., Mootha,V., Mukherjee,S., Ebert,B., Gillette,M., Paulovich,A., Pomeroy,S., Golub,T., Lander,E. and Mesirov,J. (2005) Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci USA*.

Sweet-Cordero,A., Mukherjee,S., Subramanian,A., You,H., Roix,J., Ladd-Acosta,C., Mesirov,J., Golub,T. and Jacks,T. (2005) An oncogenic KRAS2 expression signature identified by cross-species gene-expression analysis. *Nat Genet*, **37**, 48–55.

Tomfohr,J., Lu,J. and Kepler,T. (2005) Pathway level analysis of gene expression using singular value decomposition. *BMC Bioinformatics*, **6**.

Vapnik,V. (1998) *Statistical learning theory*. J. Wiley and Sons.

Warnat,P., Eils,R., and Brors,B. (2005) Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, **6**.

Welsh,M., Sapinoso,L., Su,A., Kern,S., Wang-Rodriguez,J., Moskaluk,C., Frierson,H.J. and Hampton,G. (2001) Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Res*., **61**, 5974–5978.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e117–e123
doi:10.1093/bioinformatics/btl260

Dense subgraph computation via stochastic search: application to detect transcriptional modules

Logan Everett[1], Li-San Wang[2] and Sridhar Hannenhalli*

[1]Penn Center for Bioinformatics and [2]Department of Biology, University of Pennsylvania, Philadelphia, PA, USA 19104

ABSTRACT

Motivation: In a tri-partite biological network of transcription factors, their putative target genes, and the tissues in which the target genes are differentially expressed, a tightly inter-connected (dense) subgraph may reveal knowledge about tissue specific transcription regulation mediated by a specific set of transcription factors—a tissue-specific *transcriptional module*. This is just one context in which an efficient computation of dense subgraphs in a multi-partite graph is needed.

Result: Here we report a generic stochastic search based method to compute dense subgraphs in a graph with an arbitrary number of partitions and an arbitrary connectivity among the partitions. We then use the tool to explore tissue-specific transcriptional regulation in the human genome. We validate our findings in Skeletal muscle based on literature. We could accurately deduce biological processes for transcription factors *via* the tri-partite clusters of transcription factors, genes, and the functional annotation of genes. Additionally, we propose a few previously unknown TF-pathway associations and tissue-specific roles for certain pathways. Finally, our combined analysis of Cardiac, Skeletal, and Smooth muscle data recapitulates the evolutionary relationship among the three tissues.

Contact: sridharh@pcbi.upenn.edu

1 INTRODUCTION

Eukaryotic protein coding genes are transcribed by RNA *Polymerase-II*. To accomplish this, Pol-II is critically aided by several other transcription factors (*TF*) (Kadonaga, 2004). These TFs bind to specific DNA elements in the relative vicinity of the gene, and through cooperative interaction guide Pol-II to the transcription start site (TSS). An important long-term goal is the knowledge of groups of functionally interacting factors—*transcriptional module* (Bolouri *et al*., 2002; Thompson *et al*., 2004). Transcription modules provide an efficient mechanism to co-regulate a group of functionally related genes, for instance, specific to a tissue (Wasserman *et al*., 1998) or involved in immunity (Senger *et al*., 2004).

A combinatorial approach to transcriptional module detection uses a graph-theoretical abstraction: in a bi-partite graph of TFs and genes, where a TF is connected to its target genes, a large bi-partite clique represents a potential transcriptional module (Hannenhalli *et al*., 2003). This is precisely the problem of *clique enumeration in bi-partite graphs* (Alexe *et al*., 2000). One can attach weights to the TF-gene pairs indicating the likelihood that the TF regulates the gene. In this case a more desirable optimization is to detect *heavy* sub-graphs (Tanay *et al*., 2004). These combinatorial, enumerative approaches although effective in several biological problems (Hannenhalli *et al*., 2003), are inherently inefficient, thus limiting their application. Also, a practical extension of this abstraction should include additional types of nodes in the graph, for instance functional classes or tissues. A maximal clique in a tri-partite graph with Tissue as the additional partition would reveal tissue specific transcriptional modules. One can imagine the utility of having additional partitions representing other kinds of functional information.

Efficient algebraic approaches based on spectral graph theory have been proposed to co-cluster the two dimensional gene-expression (Ernst *et al*., 2002), and word-document (Dhillon, 2001) datasets; dense blocks in the permuted matrix represent co-clusters. The main limitation with this approach is that the co-clusters are non-overlapping and it is difficult to assess their significance. Dense sub-graph computation in general graphs has been studied in the context of identifying web communities (Flake *et al*., 2000) using network flow techniques. However, these methods focus on detecting a single most dense subgraph and are not adaptable to our specific problem domain, as will become clear later. There are approaches to detect overlapping clusters, although only in 2-dimensional, gene-expression data (Ihmels *et al*., 2002).

Another desirable feature that is lacking in current approaches is that they do not distinguish a ubiquitously connected vertex from a vertex that is highly connected to a specific subset of vertices. In our application, we would like to avoid such ubiquitously connected vertices without having to filter them out in a pre-processing step. For instance a TF like Sp1 is not interesting, unless it is more tightly connected to our genes of interest than to other genes.

Here we propose a stochastic search based approach to detect dense subgraphs while addressing the concerns discussed above. We assess the significance of our solutions based on graph randomization. Our current implementation exploits the tri-partite graph structure with an arbitrary connectivity between partitions. We have applied the tool on human whole genome TF-Gene graphs for tissue specific genes to discover tissue specific transcriptional modules. We have validated the clusters detected in Skeletal muscle based

*To whom correspondence should be addressed.

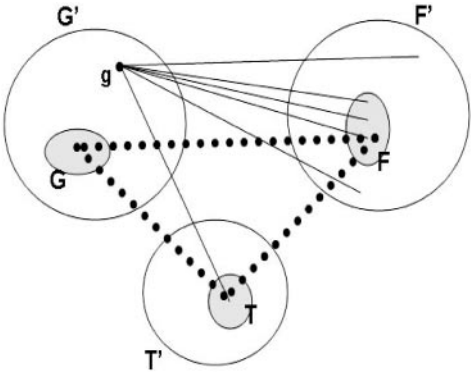


Fig. 1. Illustration of an iteration of the stochastic search.

on the literature evidence. When applied to TF-Gene-GO graph without the TF-GO edges, our approach can successfully deduce TF-GO relations, i.e. functional assignment of TFs. Similar application to TF-Gene-Pathway reveals novel TF-Pathway relations. Application to Tissue-Gene-Pathway graph using the combined datasets for Cardiac, Skeletal, and Smooth muscle recapitulates the evolutionary relationship among the three tissues and reveals novel Tissue-Pathway relations. Thus, our work presents a novel efficient approach for dense subgraphs and its application to a variety of genome wide tri-partite graphs.

2 RESULTS

2.1 Computing dense sub-graphs by Random Search—method overview

The goal of our approach is to find 'all' distinct dense sub-graphs. Because our input involves thousands of nodes and edges, our method has to be time-efficient. We adopt a *stochastic hill-climbing* approach that attempts to strike a balance between speed and premature stopping at local optima. In summary, consider a Markov chain where each state represents a potential solution (represented by an indicator variable for each node where a value of 1 indicates that the vertex belongs to the cluster). We connect each state to another state if they differ in exactly one vertex, and define the transition probability to capture the fitness of the solution. Starting from some starting state we stochastically traverse the neighborhood of this state in the state space until an optimal state is reached. We repeat this process starting from a large set of seed states to obtain several good solutions.

Although our approach is applicable to a general graph, in order to highlight the specific application to transcriptional module detection, here we illustrate the method using a tri-partite graph (Fig. 1). Let the three parts be G′ (genes), T′ (Tissues) and F′ (Transcription factors). We will also refer to these parts as G_G, G_T, and G_F respectively. Figure 1 shows the input graphs G′, T′, and F′ and a potential solution G, T, F. Intuitively, we want a solution such that nodes in G are connected to a large fraction of nodes in F and a relatively smaller fraction of nodes in F′ (same holds for all pairs of subsets). This can be captured using a log-likelihood score.

For a node g and a subset of nodes X in another partition, $N(g,X)$ is the number of nodes in X connected to g and $D(g,X) = N(g,X)/|X|$, i.e. the fraction of nodes in X that g is connected to.

The 'score' of a solution G, T, F is

$$S(G,T,F) = \sum_{u \in G \cup T \cup F} \left[\sum_{X=\{G,T,F\},\, u \notin X} N(u,X) \log \frac{D(u,X)}{D(u,X')} \right]$$

In other words, for every node, we compute its log-likelihood score with respect to each of the other partitions. In a given iteration of our stochastic search (state transition in the Markov chain), the solution can grow or shrink. Every node, both, inside and outside the current solution, is scored. The score of a node outside the current module, i.e. $g \notin G$ AND $g \in G'$, is $S(G \cup \{g\}, F, T) - S(G, F, T)$, i.e. the relative increase in the cluster score if g is *added* to the module. A node inside the module, i.e. $g \in G$ can be scored analogously as the relative increase in the module score if g is *removed*. The scores for all nodes from all partitions are normalized to a sum of 1 (after initializing the negative scores to 0). A candidate node is chosen according to this probability distribution. Note that adding or removing a single node corresponds to a state transition in our Markov chain. The procedure stops when no significant gains are achieved for several consecutive iterations.

To seed our stochastic search, we enumerate all maximal completely connected clusters with a user specified minimum number of nodes from each partition. For instance a typical value we have used is 3 genes and 3 transcription factors. We then iterate until we exhaust all seeds or reach the specified number of clusters; we choose the largest of the unused seeds and run the stochastic search algorithm to obtain a dense subgraph X; we then prune all seeds that highly overlap X to avoid finding similar subgraphs in subsequent runs. We stop after a pre-specified number of clusters (100) are identified.

Data preparation. From among the 546 vertebrate TF positional weight matrices (PWM) in TRANSFAC v8.4 (Wingender *et al*., 1996), we have extracted 221 representative PWMs (methods). This was done to minimize the bias in our clusters caused by highly similar PWMs connected to the same set of genes. For these 221 PWMs, we obtained the TF-Gene edges for all human genes using our binding site prediction method based on *Phylogenetic Footprinting* (Levy *et al*., 2002) (methods). We defined Gene-Tissue edges using an entropy-based measure of tissue-specificity (Schug *et al*., 2005) and the Novartis tissue survey data (Su *et al*., 2004). Finally Gene-GO and Gene-Pathway edges were defined using GO (Harris *et al*., 2004) and KEGG pathway resources (Kanehisa *et al*., 2002).

2.2 Tissue-specific transcriptional modules in Human—Skeletal Muscle as a case study

We identified 477 genes specifically expressed in Skeletal Muscle based on our threshold for tissue-specificity. We then applied our tool to the bi-partite graph consisting of 477 genes and 221 representative transcription factors. Figure 2 ('o') shows the cluster score distributions for this graph.

Significance To estimate the significance of the cluster scores, we randomized the input graph and computed the cluster scores, as shown in Figure 2 ('+'). A majority of the identified clusters have a score greater than the maximum score in the randomized graph. To obtain a more stringent background, we randomized the graph 100 times and for each randomized graph we retained only the maximum cluster score after running our tool until exhaustion

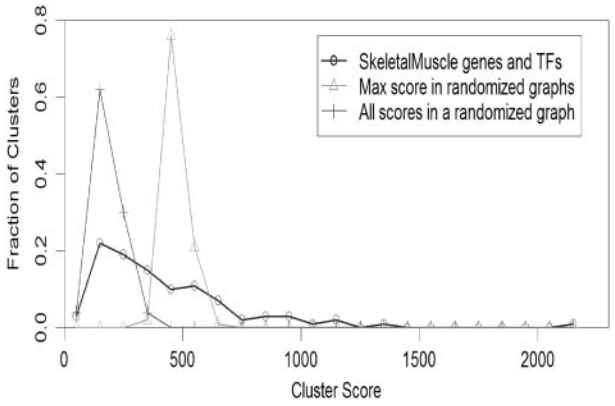


Fig. 2. The cluster score distributions for (i) clusters in Skeletal muscle specific genes and the corresponding TFs, (ii) maximum cluster score, one per randomized graph for 100 randomizations, (iii) all cluster scores for one randomized graph.

(using the same parameters as that for the original graph). As shown in the figure ('Δ'), even though the peak of real scores is to the left of the peak of max scores in the randomized graphs (this is expected since we are using max scores for the randomized graphs), there are several clusters in the input graph which score better than the maximum for any randomized graph (∼700). These 24 clusters therefore represent highly significant clusters.

Sensitivity Wasserman and Fickett have analyzed six transcription factors believed to confer muscle specific regulation (Wasserman *et al*., 1998), namely, Sp1, AP-1, Myf, SRF, MEF-2, and TEF-1. Sp1 is included in several of the top 24 clusters, including the top scoring one. In our initial grouping of positional weight matrices AP-1 belonged to a group with CREB as the representative and CREB was included among the top 24 clusters. Myf is an E-box protein and was grouped along with several other E-box proteins, with E47 as the representative, and E47 was included among the top 24 clusters. SRF was not included among the top 24 clusters but was in a cluster ranked 28, whose score is still above 90% of the background scores. MEF-2 was included in a very low scoring cluster (ranked 57). TEF-1 has a short 6 base pair binding site with very little information content, as reported in TRANSFAC and hence was not part of our input graph. However, TEF-1 is very similar to Tax/CREB in terms of binding site similarity which is in the same group as AP-1 mentioned above. Hence most of the factors analyzed in (Wasserman *et al*., 1998) are included in the high scoring clusters that we have identified.

Specificity To evaluate other factors identified by virtue of belonging to high scoring clusters, we extracted the 13 transcription factors that were included in greater than 10 of the 20 top-scoring clusters. These factors are: Sp1, MAZ, MAZR, Muscle_initiator, ETF, Churchill, EGR-1, AP2, VDR, MTF-1, Zic1, ZF5 and Spz1. MAZ (consensus: GGGGAGGG), MAZR (consensus: GGGGGGGGGGCCA), Churchill (consensus: CGGGGG) and ETF (consensus: GCGGCGG) are very similar to Sp1. ETF is a close homolog of TEF-1 (mentioned above), whereas MAZ sites are experimentally known to bind Sp1 (Parks *et al*., 1996), and MAZ is expressed in Skeletal Muscle (Song *et al*., 1998), MAZR binding site was found to be significantly enriched in 400 bp upstream of muscle genes in an independent computational analysis (Aerts *et al*., 2003). Muscle_initiator was derived by analyzing the promoters of specific Myc targets *in vivo* (Grandori *et al*., 1997). EGR-1 with SRF and Sp1 regulates muscle contraction (Irrcher *et al*., 2004). AP-2 with Sp1 regulates the muscle gene Utrophin (Perkins *et al*., 2001). VDR is involved in muscle development (Endo *et al*., 2003). MTF-1 is involved in oxidative stress response (Wimmer *et al*., 2005), an essential process in muscle. Zic1 is involved in skeletal development (Aruga *et al*., 1999). ZF5 is known to repress c-Myc (a gene involved in myogenesis) and one of the ZF5 isoforms is specifically expressed in skeletal muscle (Numoto *et al*., 1997). Thus, apart from Spz1, there is varying degree of support that all other transcription factors frequently found in high scoring clusters are involved in Skeletal muscle processes.

Although we have discussed the results only for Skeletal Muscle, we have in fact applied the tool to all tissues in the Novartis set. The score distributions follow a similar pattern relative to randomized graphs but specific analysis of the results in these tissues was not done.

2.3 Functional annotation of TFs via tri-partite cluster detection

Here we illustrate the utility of extending the above approach to multi-partite graphs. The largest cluster in the TF-Gene graphs for Skeletal muscle specific genes includes 36 TFs and 89 genes. We constructed a tri-partite graph by including the GO biological process (GOBP) for the genes as the third partition and connecting this new partition to the 'Gene' partition only. We computed dense clusters in this graph, with minimum edge density threshold of 0.75. This resulted in 14 sub-clusters. As in section 2.2, the scores of these 14 clusters are higher than the maximum scores for 100 randomized graphs (Wilcoxon rank sum test base p-value = 3.8E-04). Although, the GO annotations in these sub-clusters are largely overlapping, the genes and TFs in the sub-clusters are not so. Nevertheless it is difficult to interpret such subtly distinct sub-clusters based on the current literature. Instead, we assessed whether we can accurately assign functions to TFs via their sub-cluster membership. Recall that we did not use any known TF-GO relationships in identifying the clusters. The 14 sub-clusters involved 21 TFs and 12 GOBPs. Thus a total of 252 TF-GOBP relations are possible. In this universe of 252 relations, 59 are directly supported by the GO annotation for the TF protein, and thus represent the positives. To predict TF-GOBP relationships, we assigned each TF in a sub-cluster to each BP in that sub-cluster, resulting in 93 predicted TF-GOBP relations. Of the total of 252 relations, the overlap between predicted 93 and known 59 relations is 33 (Hypergeometric p-value = 5.4E-04). In other words 35% of our predictions include 56% of the known relations. To evaluate the validity of the 60 predicted relations with no supporting GO annotation, we took an indirect approach. For TF x and GOBP p, we estimated the *support* for a TF-GOBP relation '$x \leftrightarrow p$', as the number of $x \leftrightarrow g \leftrightarrow p$ triplets where the $x \leftrightarrow g$ indicates a binding site for x in g's promoter, and $g \leftrightarrow p$ indicates a GOBP annotation of g as p. We expect the 60 predicted TF-GOBP relations to have a greater support than the background. For the background we used the 133 of the 252 relations which were neither predicted, nor known. Also to

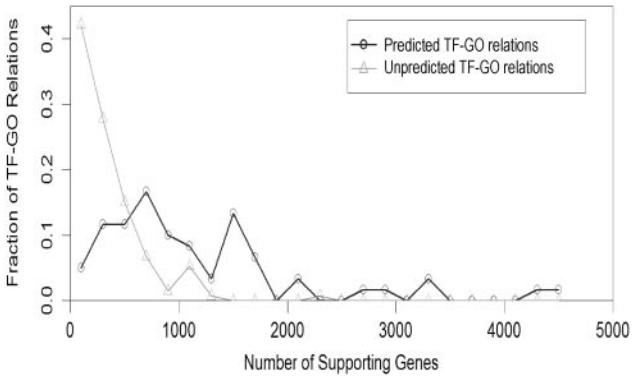


Fig. 3. Amount of indirect support for predicted TF-GOBP relations and for the background.

avoid circularity, we only used the support from genes which were *not* specific to Skeletal muscle and hence were not part of the input graph. Figure 3 shows that the predicted relations have a significantly greater support than the background (Wilcoxon rank sum test based p-value = 5.7E-15).

2.4 Co-regulation of genes involved in specific pathway

To detect specific pathways within the skeletal muscle specific modules, similar to the previous section, we constructed a tri-partite graph by including the known pathways for the genes as the third partition (instead of GO) and connecting this new pathway partition to the gene partition only. We computed 4 tightly connected clusters in this graph. One of them was intriguing in that the transcription factor ETF uniquely belonged to this sub-cluster. Recall that ETF was detected as a frequent member of high scoring clusters and is a family member of TEF-1 known to be involved in muscle processes but ETF does not have any direct evidence for involvement in muscle processes. The other TFs in this sub-cluster are p300, Sp1, AP-2 and EGR. And the genes in this sub-cluster are *Keratin 17*, *Vitronectin, Integrin-α7, Integrin-β1A,* and *cytosolic*, *malic enzyme 1*. Furthermore, the pathway 'ECM (Extra Cellular Matrix) receptor' belongs uniquely to this sub-cluster. Indeed *Vitronectin*, *Integrin-α7,* and *Integrin-β1A* belong to this pathway. ETF binding site occurs within 85 bps of a Sp1 site in the 1 kb promoter region of 4 of the 5 genes and in *Vitronectin* and *Integrin-β1A*, there are 2 distinct binding sites for ETF. Even though there is no direct experimental evidence supporting the role of ETF in the ECM-receptor pathway, we believe that the strong circumstantial evidence makes it a promising candidate to pursue for direct functional validation. Discoveries like this one can be made readily by an approach like ours that takes into account multiple types of information in an unbiased way.

2.5 Delineating Tissue-specific transcription factors and pathways via tri-partite clustering

Next we evaluate whether our approach can reveal subtle differences between tissues related at a gross level. We combined the TF-Gene data for genes specific to Heart, Skeletal muscle and Smooth muscle, resulting in a tri-partite graph consisting of 221 TFs, 1519 genes, and 3 tissues. Among the three tissues, Heart (574 genes), Skeletal muscle (477 genes), and Smooth muscle (666) genes, there are 117 genes in common between Heart and Skeletal muscle, 65 genes between Heart and Smooth muscle and 32 between Smooth and Skeletal muscle. This is consistent with phylogeny based results in (OOta *et al*., 1999). Even though cardiac muscle is evolutionarily closer to skeletal muscle, it is functionally closer to smooth muscle in that both cardiac and smooth muscle are involuntary. We investigated whether this evolutionary relationship is also reflected in the transcriptional modules. First, among the top 10 clusters, only 1 involved a single tissue and the other 9 involved exactly 2 tissues. Of these 9 cases, 6 involved Heart and Skeletal muscle, where 3 involved Heart and Smooth muscle. Thus there are twice as many clusters associating heart with skeletal muscle relative to smooth muscle. Second, among the top 10 tri-clusters, we recorded whether a TF belonged in a cluster with a tissue. For the three tissues in the specified order (Heart, Skeletal, Smooth), we assigned 3 binary numbers to each TF. For instance (1,0,1) means that the TF is associated with Heart and Smooth muscle but never with Skeletal muscle. The number of TFs belonging to the 7 possible binary vectors are—001(5), 010(0), 100(0), 110(29), 101(17), 011(0), 111(44). Thus most TFs are associated with all three tissues. Additionally there are more TFs uniquely associated with Heart and Skeletal muscle (29) than there are uniquely associated with Heart and Smooth muscle (17). Thus the transcriptional modules reflect the greater similarity between Heart and the Skeletal muscle. However, the statistical significance of this is not clear given the greater similarity between Heart and Skeletal muscle in terms of common genes.

Next we computed tri-clusters in the Tissue-Gene-Pathways graph in order to detect associations between tissues and pathways. A total of 15 clusters were detected, each with 2 tissues (this is because we required the seeds to have at least 2 tissues). Heart and Skeletal muscle co-associate in 8 cases, Heart and Smooth muscle co-associate in 5 cases and Smooth and Skeletal muscle co-associate in 2 cases. Furthermore, there are pathways that uniquely associate with one of the three tissues in our dataset. For instance there are 9 pathways uniquely associating with Heart and several of these have to do with immune system, e.g. *B cell receptor signaling pathway*, *Natural killer cell mediated cytotoxicity*, and *T cell receptor signaling pathway*. *Carbon fixation* pathway is uniquely associated with Skeletal muscle, and there are several pathways that uniquely associate with Smooth muscle, an overwhelming majority of which are involved in amino acid metabolism and degradation. We could not however assess the significance of these findings based on the current literature.

3 METHODS

Binding site annotation

We extracted the 1 kb regions upstream of the annotated transcripts in the hg16 release of the human genome from UCSC database (genome.ucsc.edu). We also extracted the Human-Mouse alignments for these regions. We searched the 1 kb regions using 546 binding profiles (Positional Weight Matrix or PWM) for vertebrate transcription factors from TRANSFAC v8.4 (Wingender *et al*., 1996). The search was done using the tool PWMSCAN (Levy *et al*., 2002). The initial hits were based on a p-value cutoff of 0.0002, corresponding to an average frequency of 1 hit every 5 kb scanned in the human genomic background. We filtered these initial

hits further using Human-Mouse alignments. For each hit we computed the fraction c of binding site bases that were identical between human and mouse. We retained the hits such that either p-value $\leq$ 0.00002 (1 in 50 kb) or $c \geq 0.8$. This procedure is similar to the one reported previously (Levy *et al*., 2002).

Clustering transcription factor PWMs

Pair-wise similarity computation Each PWM X is a 4 by k matrix for k-length binding site, where X_{ui} is the proportion of base u at position i, such that $\sum_u X_{ui} = 1$ (Stormo, 2000). We compute the dissimilarity or distance between position i of PWM X and position j of PWM Y using relative entropy $RE_{ij} = \sum_u X_{ui} \ln (X_{ui}/Y_{uj})$ (Durbin *et al*., 1998). For two identical positions this value is 0 and the more dissimilar the positions, the higher the RE value. However, as defined, this is an asymmetric measure and in practice we take the average of R_{ij} and R_{ji} as the distance between the two positions. Notice that according to this measure, for two positions at which the base pairs are distributed according to the background probability (say, equi-probable), their RE value will be 0, even though individually these positions are not informative. Let R_{ir} be the RE-value between column i and background probability distribution of bases. R_{jr} is defined similarly. We define the similarity between column i and column j, $S_{ij} = R_{ir} + R_{jr} - ((R_{ij} + R_{ji})/2)$. We first compute the S_{ij} for every pair of columns for all PWMs in the TRANSFAC database. These values are normally distributed with mean μ and standard deviation σ. The sum of k such S-values is also normally distributed with mean $\mu_k = \mu k$, and standard deviation $\sigma_k = \sigma\sqrt{k}$. To compute the similarity between k consecutive columns of two PWMs, we sum up the k S-values for aligned column pairs and transform this value to a z-score $= (S - \mu_k)/\sigma_k$, which makes the scores for different values of k comparable. Next, for every PWM-pair and for every alignment offset with a minimum of 6 base overlap between the PWMs (i.e., $k \geq 6$), we compute the similarity z-score ('z-value'). Using the empirical distribution of z-values for all alignments of all PWM pairs, we convert each individual z-value into a p-value, i.e., the probability of observing the z-value or higher in the background distribution; we call this the pz-value. Finally, to compute the similarity between two PWMs X and Y while allowing for the possibility that two related PWMs may be slightly shifted in positions, we slide the PWMs relative to each other such that at least 6 positions are aligned. For each such offset we compute the *pz*-value. Let *mpz* be the minimum *pz*-value over all offsets. Notice that the longer PWM pairs have a greater number of possible offsets and thus tend to achieve a low *mpz*-value. To correct for this effect, we compute the significance of the observed *mpz*-value as the random expectation of observing the mpz-value for K trials where K is the number of offsets. That is,

$$P(X,Y) = 1 - (1 - mpz(X,Y))^K.$$

Clustering PWMs based on the P-values Given a p-value threshold (we use 0.005), all PWMs can be represented as a network where PWMs correspond to the nodes and two nodes are connected if their similarity p-value is below the threshold. We then compute the so-called bi-connected component in this graph. A bi-connected component is a connected component of the graph that remains connected if any of the nodes are removed. Each bi-connected component corresponds to a cluster. In other words if two PWMs belong to same cluster, they must have at least two independent lines of evidence that they are related (i.e. paths in the graph). Each cluster thus obtained represents a family of PWMs with similar DNA binding specificity. We selected the median of each cluster as the cluster representative. Out of 546 PWMs, 442 were grouped into 117 clusters, and with 104 singletons, this procedure resulted in 221 representative PWMs.

Tissue specific genes

For each gene g and each tissue t, we say *g is specific for t* if its expression level in t is considerably higher than in other tissues, using the following procedure from (Schug *et al*., 2005). We use the Novartis GeneAtlas expression dataset (Su *et al*., 2004): the dataset has 79 different types of human tissues (two replicates each). The hybridization experiments are done using the Affymetrix HG-U133A (33689 probesets) and GNF1B (11391 probesets) platforms. Let $w(g,t)$ be the average expression level of probeset g in tissue t (*not* log2-transformed) over the two replicates. For each probeset, the relative expression level for tissue t is $p(t \mid g) = w(g,t)/\sum_{\text{tissue } i} w(g,i)$. The entropy of gene g is

$$H(g) = -\sum_t p(t \mid g) \log_2 p(t \mid g).$$

The categorical specificity of gene g and tissue t is $Q(g \mid t) = H(g) - \log_2 p(t \mid g)$. A low Q score implies gene g is highly specific for tissue t: $H(g)$ is low when the expression level of g is concentrated in a few tissues, whereas $p(t \mid g)$ is high when g is highly expressed in t. We empirically chose a value of 10.5 as the cutoff for $Q(g \mid t)$, as the density of the gene-tissue specificity begins a sharp increase at a higher Q. A more stringent value of 7 was suggested in (Schug *et al*., 2005). We then remap the association from Affymetrix probeset IDs to RefSeq IDs.

KEGG and GO annotation data

We built the associations between genes (refseq ID), KEGG pathways, and GO terms as follows. We downloaded data from the KEGG server that contained the association data between KEGG pathways and NCBI GI numbers. We downloaded the association data between GO terms and NCBI GENE IDs from the NCBI server. The mappings from GI numbers and GENE IDs to RefSeq IDs are obtained from NCBI. The mapping is inclusive: for example, if KEGG pathway x is associated with GI number y, and y is mapped to RefSeq IDs a, b, and c, then x is associated with a, b, and c.

Graph randomization

To determine the significance of cluster scores we find clusters by an identical process on randomized graphs with the node degrees identical to the real graph. The graph randomization process is performed by swapping edges with non-edges under a condition that preserves the degrees of all nodes. Specifically, a quadruple of nodes (w,x,y,z) qualifies for this swapping condition if it meets the following criteria (Yeger-Lotem *et al*., 2004): (i) both w and x reside in the same partition A, and both y and z reside in another partition B; (ii) there exists an edge between w and y, denoted as $E(w, y) = 1$, and also an edge between x and z, denoted as $E(x, z) = 1$; and 3. There exists a non-edge between w and z, and a non-edge between x and y, denoted as $E(w,z) = 0$ and $E(x,y) = 0$ respectively. If the quadruple of nodes meets these criteria, we then swap the edges by setting $E(w,y) = E(x,z) = 0$ and $E(w,z) = E(x,y) = 1$.

We sufficiently randomize an edge set between two partitions by selecting a pair of nodes from each of the two partitions at random and swapping the edges between these nodes if the above criteria are satisfied. This process is repeated until the number of successful swaps is twice the total number of edges. The number of swaps required to sufficiently randomize a graph was determined by measuring the hamming distance from a representative graph after each swap operation was performed.

4 DISCUSSION

The problem of efficient computation of tightly connected clusters in a network has been studied in several biological as well as non-biological contexts. As we have argued, however, the current approaches are either (i) computationally inefficient, (ii) detect *one optimal* cluster, (iii) find a few *disjoint* bi-clusters, or (iv) do not discriminate against ubiquitously connected nodes. The trivial approach to mask the best solution and repeat the process to find other solutions leaves us with the problem of finding the best way to mask current clusters and is not at all obvious. All previous applications in biology are limited to two partitions, typically

genes and expression conditions and there remains a need to extend this to multiple partitions. Our emphasis has been on developing an adaptable and general approach to finding meaningful clusters in a collection of interrelated heterogeneous datasets.

The problem of identifying dense subgraphs in a general graph (not necessarily a multipartite graph) has been studied in other contexts using combinatorial approaches. These approaches aim at finding the optimum (densest) subgraph. One can model this problem in a way that is amenable to a *Monte Carlo Markov Chain* (*MCMC*) technique, like *Gibbs sampling*. Briefly, we can model the edges in the graph as being generated by two distinct probability distributions depending on whether the edge belongs to the (unknown) dense subgraph or not. The unknown parameters including the edge probabilities and the cluster membership can be iteratively estimated. In fact one can also design an *Expectation Maximization* (*EM*) using the above setup. Although we have modeled the problem as a Markov chain, we have decided to search for a locally optimal cluster using a stochastic hill-climbing approach. The main reason for this is the adaptability/generality of the approach to a graph with arbitrary number of partitions and arbitrary connectivity. Any given problem domain entails different types of entities (partitions) with a different level of connectivity between partitions. It was thus important to design the method in a configurable fashion and our particular approach allows that. We have not discussed, due to lack of space, the various configuration parameters that our current implementation allows. For instance, in principle, we can have a specific schedule for selecting edges from different partitions to influence the detected clusters if we had an *a priori* knowledge. Our cluster score can be easily extended to weight edges or weight partitions and this kind of adaptability is difficult to achieve with a more standard approach like EM or Gibbs sampling. Our current implementation is 'work in progress' and this work illustrates the utility of such a tool. A fully configurable tool for finding dense subgraphs will be published in future work.

Efficient generation of seeds presents the computational bottleneck. We have followed a simple enumerative approach, given the seed size relative to different partitions. For a seed size of k in one of the partitions, we enumerate all k-vertex sets in that partition and look for neighboring vertices in other partitions in search of a seed above a specified size. This can become prohibitive for a partition with several hundred vertices and $k > 4$. By carefully choosing the partition to enumerate over, we have tried to counter this problem to some extent.

There are very few examples of experimentally determined transcriptional modules, thus making a large-scale evaluation of computational methods difficult. However, we have shown using a variety of validation approaches, that (i) the cluster scores are highly significant, (ii) we can detect almost all of the established TFs involved in Skeletal muscle specific expression, (iii) almost all of the highly frequent TFs have literature evidence for involvement in Skeletal muscle gene regulation, (iv) using a TF-Gene-GO graph, we can successfully assign function to TFs, (v) in a combined set of 3 tissues, the detected transcriptional modules support evolutionary relationship between Cardiac, Skeletal and Smooth muscle, and (vi) novel hypotheses regarding TF-Pathway and Tissue-Pathway can be generated using our approach.

Besides applying our tool to additional datasets, our future plan includes (i) Extensive simulation studies and incorporation of other score functions that account for edge weights, and (ii) extending the current implementation to a graph with arbitrary number of partitions.

ACKNOWLEDGEMENTS

L.W. was supported by an NIH postdoctoral training grant in Computational Biology Authors wish to thank Larry Singh for his comments on the manuscript. S.H. was supported by the University of Pennsylvania startup funds.

REFERENCES

Aerts,S., Thijs,G., Coessens,B., Staes,M., Moreau,Y. *et al.* (2003) 'Toucan: deciphering the cis-regulatory logic of coregulated genes'. *Nucleic Acids Res.*, **31** (6), 1753–64.

Alexe,G., Alexe,S., Foldes,S. and Hammer,P. (2000) Consensus algorithm for generation of all maximal bi-cliques. DIMACS tech report: 1–20.

Aruga,J., Mizugishi,K., Koseki,H., Imai,K., Balling,R. *et al.* (1999) 'Zic1 regulates the patterning of vertebral arches in cooperation with Gli3'. *Mech. Dev.*, **89** (1–2), 141–50.

Bolouri,H. and Davidson,E. H. (2002) 'Modeling DNA sequence-based cis-regulatory gene networks'. *Dev. Biol.*, **246** (1), 2–13.

Dhillon,I. S. (2001) *Co-clustering documents and words using bipartite spectral graph partitioning*. Knowledge Discovery and Data Mining.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis*. Cambridge University Press, Cambridge, UK.

Endo,I., Inoue,D., Mitsui,T., Umaki,Y., Akaike,M. *et al.* (2003) 'Deletion of vitamin D receptor gene in mice results in abnormal skeletal muscle development with deregulated expression of myoregulatory transcription factors'. *Endocrinology*, **144** (12), 5138–44.

Ernst,J., Heun,V. and Voll,U. (2002) *Generalized Clustering of Gene Expression Profiles—A Spectral Approach*. International Conference of Bioinformatics, INCOB'02, Bangkok, Thailand.

Flake,G. W., Lawrence,S., Giles,C. L. and Coetzee,F. (2000) 'Self-Organization of the Web and Identification of Communities'. *IEEE Computer*, **35** (3), 66–71.

Grandori,C. and Eisenman,R. N. (1997) 'Myc target genes'. *Trends. Biochem. Sci.*, **22** (5), 177–81.

Hannenhalli,S. and Levy,S. (2003) 'Transcriptional regulation of protein complexes and biological pathways'. *Mamm Genome*, **14** (9), 611–9.

Harris,M. A., Clark,J., Ireland,A., Lomax,J., Ashburner,M. *et al.* (2004) 'The Gene Ontology (GO) database and informatics resource'. *Nucleic Acids Res.*, **32** (Database issue), D258–61.

Ihmels,J., Friedlander,G., Bergmann,S., Sarig,O., Ziv,Y. *et al.* (2002) 'Revealing modular organization in the yeast transcriptional network'. *Nat. Genet.*, **31** (4), 370–7.

Irrcher,I. and Hood,D. A. (2004) 'Regulation of Egr-1, SRF, and Sp1 mRNA expression in contracting skeletal muscle cells'. *J. Appl. Physiol.*, **97** (6), 2207–13.

Kadonaga,J. T. (2004) 'Regulation of RNA polymerase II transcription by sequence-specific DNA binding factors'. *Cell*, **116** (2), 247–57.

Kanehisa,M., Goto,S., Kawashima,S. and Nakaya,A. (2002) 'The KEGG databases at GenomeNet'. *Nucleic Acids Res.*, **30** (1), 42–6.

Levy,S. and Hannenhalli,S. (2002) 'Identification of transcription factor binding sites in the human genome sequence'. *Mamm. Genome*, **13** (9), 510–4.

Numoto,M., Yokoro,K., Yasuda,S., Yanagihara,K. and Niwa,O. (1997) 'Detection of mouse skeletal muscle-specific product, which includes ZF5 zinc fingers and a VP16 acidic domain, by reverse transcriptase PCR'. *Biochem. Biophys. Res. Commun.*, **236** (1), 20–5.

OOta,S. and Saitou,N. (1999) 'Phylogenetic relationship of muscle tissues deduced from superimposition of gene trees'. *Mol. Biol. Evol.*, **16** (6), 856–67.

Parks,C. L. and Shenk,T. (1996) 'The serotonin 1a receptor gene contains a TATA-less promoter that responds to MAZ and Sp1'. *J. Biol. Chem.*, **271** (8), 4417–30.

Perkins,K. J., Burton,E. A. and Davies,K. E. (2001) 'The role of basal and myogenic factors in the transcriptional activation of utrophin promoter A: implications for therapeutic up-regulation in Duchenne muscular dystrophy'. *Nucleic Acids Res.*, **29** (23), 4843–50.

Schug,J., Schuller,W. P., Kappen,C., Salbaum,J. M., Bucan,M. *et al.* (2005) 'Promoter features related to tissue specificity as measured by Shannon entropy'. *Genome Biol.*, **6** (4), R33.

Senger,K., Armstrong,G. W., Rowell,W. J., Kwan,J. M., Markstein,M. *et al.* (2004) 'Immunity regulatory DNAs share common organizational features in Drosophila'. *Mol. Cell*, **13** (1), 19–32.

Song,J., Murakami,H., Tsutsui,H., Tang,X., Matsumura,M. *et al.* (1998) 'Genomic organization and expression of a human gene for Myc-associated zinc finger protein (MAZ)'. *J. Biol. Chem.*, **273** (32), 20603–14.

Stormo,G. D. (2000) 'DNA binding sites: representation and discovery'. *Bioinformatics*, **16** (1), 16–23.

Su,A. I., Wiltshire,T., Batalov,S., Lapp,H., Ching,K. A. *et al.* (2004) 'A gene atlas of the mouse and human protein-encoding transcriptomes'. *Proc. Natl Acad Sci. USA*, **101** (16), 6062–7.

Tanay,A., Sharan,R., Kupiec,M. and Shamir,R. (2004) 'Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data'. *Proc. Natl Acad Sci. USA*, **101** (9), 2981–6.

Thompson,W., Palumbo,M. J., Wasserman,W. W., Liu,J. S. and Lawrence,C. E. (2004) 'Decoding human regulatory circuits'. *Genome Res.*, **14** (10), 1967–74.

Wasserman,W. W. and Fickett,J. W. (1998) 'Identification of regulatory regions which confer muscle-specific gene expression'. *J. Mol. Biol.*, **278** (1), 167–81.

Wimmer,U., Wang,Y., Georgiev,O. and Schaffner,W. (2005) 'Two major branches of anti-cadmium defense in the mouse: MTF-1/metallothioneins and glutathione'. *Nucleic Acids Res.*, **33** (18), 5715–27.

Wingender,E., Dietze,P., Karas,H. and Knuppel,R. (1996) 'TRANSFAC: a database on transcription factors and their DNA binding sites'. *Nucleic Acids Res.*, **24** (1), 238–41.

Yeger-Lotem,E., Sattath,S., Kashtan,N., Itzkovitz,S., Milo,R. *et al.* (2004) 'Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction'. *Proc. Natl Acad Sci. USA*, **101** (16), 5934–9.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e124–e131
doi:10.1093/bioinformatics/btl210

Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle

Adrien Fauré, Aurélien Naldi, Claudine Chaouiya and Denis Thieffry*

Institut de Biologie du Développement de Marseille-Luminy, Campus scientifique de Luminy, CNRS case 907, 13288 Marseille, France

ABSTRACT

Motivation: To understand the behaviour of complex biological regulatory networks, a proper integration of molecular data into a full-fledge formal dynamical model is ultimately required. As most available data on regulatory interactions are qualitative, logical modelling offers an interesting framework to delineate the main dynamical properties of the underlying networks.

Results: Transposing a generic model of the core network controlling the mammalian cell cycle into the logical framework, we compare different strategies to explore its dynamical properties. In particular, we assess the respective advantages and limits of synchronous versus asynchronous updating assumptions to delineate the asymptotical behaviour of regulatory networks. Furthermore, we propose several intermediate strategies to optimize the computation of asymptotical properties depending on available knowledge.

Availability: The mammalian cell cycle model is available in a dedicated XML format (GINML) on our website, along with our logical simulation software GINsim (http://gin.univ-mrs.fr/GINsim). Higher resolution state transitions graphs are also found on this web site (Model Repository page).

Contact: thieffry@ibdm.univ-mrs.fr

1 INTRODUCTION

A proper understanding of the structure and temporal behaviour of biological regulatory networks requires the integration of regulatory data into a formal dynamical model (for a review, see de Jong, 2002). Although this issue has been recurrently addressed by applying standard mathematical approaches (*e.g.*, differential or stochastic equations) borrowed from physical sciences, it is notably complicated by the diversity and sophistication of regulatory mechanisms, as well as by the chronic lack of reliable quantitative information.

This situation has motivated the development of intrinsically qualitative approaches leaning on Boolean algebra or generalisation thereof (Glass & Kauffman, 1973; Thomas, 1991).

In this paper, we lean on previous work refining, extending and implementing the logical approach initially formulated by R. Thomas *et al*. (Thomas, 1991; Thomas *et al*., 1995). The corresponding framework is summarised in the following section (see Chaouiya *et al*., 2003, for more detail). This framework is then used to derive a logical version of the differential model for the control of the mammalian cell cycle recently published by Novak and Tyson (2004). The corresponding regulatory network is described in the last chapter of the introduction, together with citations of the most relevant experimental articles (for a didactic introduction to cell cycle modelling, see Fuß *et al*., 2005).

In their landmark model analysis, Novak and Tyson have heavily relied on numerical integration techniques (temporal simulations, phase space analyses, and bifurcation diagrams) to delineate the main dynamical properties of the complex regulatory system under study. Their results are valid for specific sets of parameter values and function shapes, which are difficult to establish quantitatively. Furthermore, such parametric analyses can only handle a few parameters at once.

In contrast, although much more qualitative, the logical framework enables a more systematic and extensive characterisation of all the behaviours compatible with a given regulatory graph. Furthermore, this framework offers enumerative or analytical means to identify relevant asymptotical behaviours (stable states, state transition cycles, etc.). Finally, extending a logical model to encompass additional regulatory modules is relatively easy.

However, one difficulty with the logical approach lies in the implicit treatment of time. In this respect, different approaches have been proposed, either considering all transitions under a synchronicity assumption, or considering them under a fully asynchronous assumption, *i.e.*, selecting a single transition at each dynamical step. The first assumption is simple but leads to well known artefacts, whereas the results obtained under the second assumption are more difficult to evaluate. In this paper, we explore the use of different strategies enabling a honourable compromise between these two extreme approaches.

1.1 Logical modelling of regulatory networks

The specification of a logical model involves three main steps: (i) the building of a *regulatory graph*; (ii) the definition of the *logical parameters* of the system; (iii) the specification of the *updating assumption*(s).

Cross-regulations between regulatory components are formalized in terms of an oriented graph. In this *regulatory graph*, the vertices represent the different regulatory components (activity of a gene, concentration of a regulatory product, or level of activity of a protein), whereas the edges represent regulatory interactions between these components (including self-regulations). The level or activity

*To whom correspondence should be addressed.

of a regulatory component *i* is given by an integer, taking its values in the interval [0, Max_i], where Max_i is the maximal value considered for this element (in the simplest, Boolean case, Max_i is set to 1). Each edge is labelled with an interval of integers defining the set of values for which the source of the interaction influences its target. Naturally, this interval must be compatible with the values allowed for the source of the interaction. Furthermore, for sake of simplification, the maximal value of the interval is usually set to the maximal value of the source of the interaction (notion of *threshold*). Note that this definition allows the specification of multiple interactions between two components, provided that each interaction involves a specific *threshold* (alternatively, disjoint contiguous intervals can be used).

Finally, an edge can also be optionally labelled with a positive or negative sign, which then specifies that the effect of the source on the target is monotonous, either potentially activating or inhibiting the target, respectively. The specification of interaction signs only affects the graphical representation and must be translated into proper parameter values to obtain coherent regulatory effects.

The next step consists in defining the combinatory effects of the regulatory inputs on the expression or activity of a given component of the regulatory graph. The set of inputs is already specified at the level of the regulatory graph. However, the effect of each regulator usually depends on the presence of the co-regulators. For the sake of conciseness, we consider only the combinations of interactions allowing a significant (non zero, from a logical point of view) expression or activity of the regulated component. The corresponding *logical parameters* are each univocally identified by the set of interactions acting on the regulated genes and take their values in [1,Max_{target}] (see the next section for a concrete illustration).

The dynamical behaviour of a logical regulatory model is represented in terms of an oriented graph, where each vertex represents a specific *logical state* of the system (*i.e.*, a vector giving the discrete levels of expression/activity of all the components), whereas the edges represent (possible) *transitions* between these states.

Together with the regulatory graph, the logical parameters define the rules directing the dynamics of a network, *i.e.*, the potential occurrence of specific edges in the *state transition graph*. Indeed, at a given state, a specific logical parameter can be associated with each component. If the value of this parameter is smaller or greater than that of the concentration/activity level of the corresponding component, this level will tend to decrease or increase, respectively. Otherwise (when the parameter value and the corresponding component level are equal), the component will tend to keep its current value.

At this stage, different assumptions might be considered. According to the simplest one, at a given state, all increase or decrease calls are realized simultaneously (*synchronous updating*), changing the component levels by one unit at a time (see *e.g.*, Kauffman, 1993). Easy to implement and computationally efficient, this approach leads to well known dynamical artefacts (in particular spurious cycles). At the other extreme of the spectrum, the transition calls can be asynchronously updated, *i.e.*, one single transition will be selected at a time. This assumption requires additional rules to sort out concurring transitions (*e.g.*, the specification of time delays or of priorities). These additional rules are tricky to define, as they may perfectly be context sensitive, *i.e.*, finely depend on the levels of various regulatory components (although these combinations might correspond to identical parameter values). For this reason, all possible transitions are often generated, and an *asynchronous transition graph* is built where all single possible transitions are considered, although all resulting dynamical pathways cannot be followed for a single set of transition rules.

Whatever the updating assumption, of particular interest is the asymptotical behaviour of the system, *e.g.*, the terminal vertices (*stable states*, with no updating calls) or the attractive cycles found in the state transition graph. Note that such *attractors* (in particular the stable states and simple terminal cycles) can easily be located in the context of the synchronous updating assumption. As we shall see, the synchronous assumption can often (but not always) be considered as a shortcut for the computation of the asynchronous dynamics. This point will be further assessed below through the analysis of a logical model of the core network controlling the mammalian cell cycle.

To ease the definition of a regulatory graph and of the associated logical parameters, as well as the construction of the (a)synchronous state transition graphs, we have developed a logical modelling/analysis/simulation software called *GINsim* (Gonzalez *et al.*, 2006). A new release of *GINsim* now implements the possibility to play with the different updating assumptions and to define different *priority classes*.

Let consider a regulatory graph with n nodes $\{g_1, g_2, \ldots g_n\}$. A logical state is a vector $S=(s_1, s_2, \ldots s_n)$ where s_i is the current level of the *i*th regulatory product ($s_i \in \{0, \ldots Max_i\}$). Given such a state, it is possible to determine the evolution of g_i, for all $i = 1, \ldots n$. Indeed, given any regulatory component g_i, the interactions which are operating on g_i in the state S can be identified, and the relevant logical parameter (*i.e.*, corresponding to the right combination of incoming interactions) gives the value k_i to which g_i should tend. If $s_i > k_i$ (the current level is greater than the parameter value), there is a call for decreasing the level of g_i, (a decrease call on g_i is denoted $g_i\downarrow$); if $s_i < k_i$, there is a call for increasing the level of g_i (denoted $g_i\uparrow$); otherwise (if $s_i = k_i$), there is no updating call for this component. A stable state is thus a state without updating call.

The synchronicity assumption amounts to apply all concurrent transitions simultaneously, all states having thus at most one successor; under the asynchronous assumption, concurrent transitions are applied separately, and a state with q updating calls has then exactly q successors.

Here, we introduce a new functionality of *GINsim* consisting in the definition of p priority classes $C_1, C_2, \ldots C_p$, *with* $p \leq n$, which gather regulatory products depending on their qualitative production and/or degradation delays:

(i) each class C_i is associated with a rank $r(C_i)$ ($1 \leq r(C_i) \leq p$, 1 being the highest rank), as well as with an updating policy (synchronous or asynchronous);

(ii) several classes may have the same rank; and concurrent transitions on genes of different classes with identical rank are triggered asynchronously;

(iii) at any state S, among all concurrent transitions, only those on genes of the classes with the highest rank are triggered;

(iv) concurrent transitions inside a class are triggered accordingly to the updating policy associated to that class;

(v) finally, increasing and decreasing transitions of each gene can be distinguished and associated to classes with different ranks.

1.2 Regulation of the mammalian cell cycle

The cell cycle involves a succession of molecular events leading to the reproduction of the genome of a cell (Synthesis or S phase) and its division into two daughter cells (Mitosis, or M phase). The M phase itself encompasses different sub-phases (prophase, metaphase, anaphase, telophase) characterised by specific chromosomal and nuclear changes (respectively: condensation of the chromatin, alignment of the chromosomes, separation of the sister chromatids, and formation of the two daughter nuclei). The S and M phases are preceded by two gap phases, called G1 and G2 respectively (for a review, see, for example, Tessema *et al*., 2004). These events are very well known and can easily be monitored with an optical microscope.

During the late 1970s and early 1980s, yeast geneticists have identified the cell-division-cycle (cdc) genes, encoding for new classes of molecules including the cyclins (so called because of their cyclic pattern of activation), and their cyclin dependent kinases (cdk) partners. Since then, our knowledge of the molecular network that controls cell cycle events has tremendously progressed, but the number of components and interactions known to be involved has so much increased that proper formal modelling becomes necessary to understand the behaviour of such a complex system.

Our model analysis is rooted in the seminal work of Novak and Tyson, who have recently derived and analyzed a set of 18 ordinary differential equations (ODE) to model the control of the restriction point of the mammalian cell cycle (Novak and Tyson, 2004). Based on this differential model and using numerical integration techniques, the authors were able to qualitatively reproduce the main known dynamical features of the wild-type biological system, as well as the consequences of several types of perturbations. This state-of-the-art model study nevertheless appears difficult to extend, although there is clearly many more regulators, variants and interactions to consider (see, *e.g*., Kohn's map at http://discover.nci.nih.gov/kohnk/fig6a.html).

In this respect, the logical formalism offers an appropriate framework to qualitatively explore the dynamical properties of relatively complex regulatory graphs. However, up to now, it has been mostly applied to transcriptional regulatory networks, and its application to the numerous and various protein interactions at the core of the cell cycle network was thus a challenge.

As a starting point, we have used Novak and Tyson's diagram and model to build a logical regulatory graph (see Figure 1). In the process, we were led to derive a proper logical representation for each type of regulatory interaction. In what follows, we summarize the main experimental data and assumptions underlying our regulatory graph. In the context of this paper, we further focus on a specific Boolean version of this model.

Mammalian cell division is tightly controlled, for it must be coordinated with the overall growth of the organism, as well as answer specific needs, such as wound healing for example. This coordination is achieved through extra-cellular positive and negative signals whose balance decides whether a cell will divide or remain in a resting state (quiescence or G0 phase), which can be reached and left by the cell during the G1 phase. The positive signals or growth factors ultimately elicit the activation of Cyclin D in the cell. In our model, CycD thus represents the input, and its activity is considered constant. Note that cdk4 and cdk6, the partners of Cyclin D, are not explicitly represented in our model, for their activity essentially depends on the presence or absence of their cyclin. In other words, CycD stands here for the whole cdk4/6-Cyclin D complex. The same approach has been adopted for the other cyclin/cdk pairs.

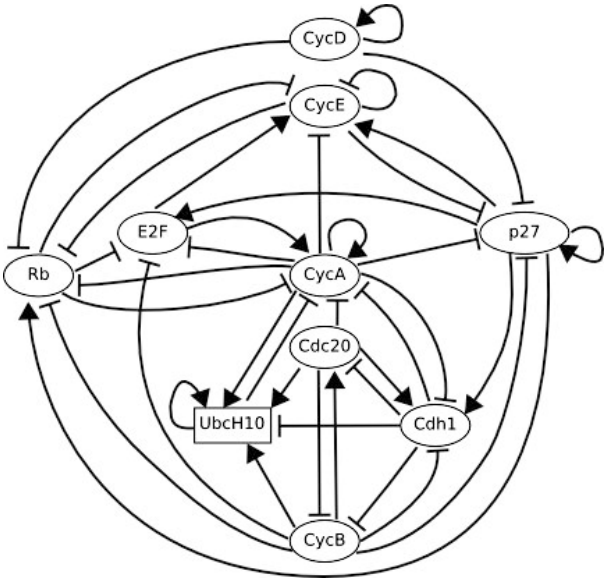


Fig. 1. Logical regulatory graph for the mammalian cell cycle network. Each node represents the activity of a key regulatory element, whereas the edges represent cross-regulations. Blunt arrows stand for inhibitory effects, normal arrows for activations.

In our model, CycD is necessary for the inactivation of the retinoblastoma protein Rb, and for the sequestration of p27/Kip1 (p27 in the sequel). This protein is a cdk inhibitor that sequesters cdk2/Cyclin E (CycE) and cdk2/Cyclin A (CycA), preventing them from phosphorylating their targets (reviewed in Coqueret, 2003). It is usually considered that Cyclin D remains active when in complex with p27, though the issue is still debated (Olashaw *et al*., 2004). For the sake of simplicity, we consider that CycD directly inhibits p27.

In contrast, the complexes formed by p27 and CycE or CycA are represented in a subtler way, though this formation remains implicit: when both p27 and CycE or CycA are active, the complex forms, and the activity of the cyclin is blocked. To model the fact that the cyclins remain present though sequestered when linked to p27, we consider that p27 opposes their activities on their targets, instead of directly inhibiting them. In our model, this is embodied by arrows from p27 onto the targets of CycE and CycA, with a sign opposite to that corresponding to the effect of the cyclins on their targets in the absence of p27.

The other target of CycD, Rb, is a key tumour suppressor, which is found mutated in a large variety of cancers. Rb is inactivated by phosphorylation, and CycD is involved in the first step of this process (reviewed in Tamrakar *et al*., 2000). However, in this simplified Boolean model, we consider that Rb inactivation by CycD is total.

Rb forms a complex with members of the E2F family of transcription factors, turning them from transcriptional activators to repressors, in part through recruitment of chromatin remodelling

complexes. For this reason, we model the action of Rb by direct inhibitions of E2F targets (which include E2F itself).

E2F is a wide family of dimeric transcription factors, formed by a member of the E2F family, and a member of the DP family. It is usually divided into activators E2Fs (E2F1, E2F2, E2F3a) and repressors E2Fs (E2F3b, E2F4, E2F5), plus the recently discovered E2F6, E2F7 and E2F8, whose structure, regulation and mode of action are slightly different from those of the *regular* E2Fs (Dimova and Dyson, 2005). In our present model, E2F represents the activator members (together with their DP partners), the other E2Fs being implicit.

At the G1/S transition, E2F activates the transcription of Cyclin E, which in turns causes the inactivation of Rb. CycE also phosphorylates p27, eliciting its destruction. Phosphorylated Rb dissociates from E2F, allowing more Cyclin E to be transcribed, further increasing the phosphorylation of Rb and the destruction of p27, in a positive feedback loop.

Cyclin A is another target of E2F, which is activated slightly after Cyclin E, when Rb is more completely inactivated (Zhang *et al*., 2000). The action of CycA contributes to maintain Rb and p27 inhibition, inactivates E2F and CycE and most importantly, inactivates the Anaphase Promoting Complex (APC).

The APC is an important E3 ubiquitin ligase that is activated in a cyclic fashion (reviewed in Harper *et al*., 2002). The APC complex is represented by its two activators, Cdh1 and Cdc20. Around the G2-to-M-phase transition, CycA inactivates Cdh1, which switches the APC OFF, allowing Cyclin B to appear. Cyclin B in turn activates Cdc20, sowing the seeds of its own destruction, since CycB is a target of Cdc20. Cdc20 is responsible for the metaphase-to-anaphase transition: it activates separase through the destruction of its inhibitor securin; this activation elicits the cleavage of the cohesin complexes that maintain the cohesion between the sister chromatids, thus leading to their separation. Cdc20 also participates in degrading CycA, and indirectly activates Cdh1. Cdh1 completes CycA and CycB inactivation, and inactivates Cdc20. In absence of its inhibitors, E2F can be reactivated and a new cycle begins.

How Cyclin A can rise a level high enough to inactivate its own inhibitor has long remained a paradox. Rape and Kirshner (2004) solved it by highlighting the role of the E2 ubiquitin conjugating enzyme UbcH10). They found that UbcH10 is necessary for Cdh1 dependent degradation of Cyclin A, but not of the other APC substrates; once all of its substrates have been degraded, UbcH10 can ubiquitinate itself, preventing the APC from degrading Cyclin A, which can thus reappear. These findings make the activation of Cyclin A in S phase coherent with the observation that Cdh1 is still active at this point of the cycle (Huang *et al*., 2001). At the present stage, the explicit inclusion of UbcH10 constitutes the most remarkable extension of Novak & Tyson's model. It further allows us to incorporate an important additional interaction, the inactivation of CycA by Cdh1 (within the APC complex).

2 RESULTS

2.1 Regulatory graph and its parameterization

The Figure 1 displays the regulatory graph integrating all the data briefly reviewed in the introductory section.

On the basis of this graph and using additional information from the literature, it is possible to derive a set of rules enabling the activation of each of the regulatory component encompassed by this graph. Presented in Table 1, these rules are sufficient to derive all the non-zero logical parameters enabling the recovery of the main known features of the wild-type cell cycle.

In its present Boolean version (*i.e*., $Max_i = 1$ for all regulatory components), our model is still simple enough to allow an exhaustive dynamical analysis with the logical simulation software *GINsim*. The complete state transition graph contains 1024 vertices (*i.e*., Boolean states). To study the dynamical trajectories corresponding to the asymptotical behaviour of the system, we still need to specify an updating assumption. As we shall see, this specification further determines the pathway(s) followed by the system, in particular with respect to the cyclic attractor.

2.2 Synchronous *versus* asynchronous updating

Starting with the simplest, synchronous assumption, we obtain two attractors. The first one is a stable state with only Rb, p27 and Cdh1 active, in the absence of CycD; this state is reached from all the other states lacking CycD activity (*i.e*., in the lack of growth factors; this state thus corresponds to the phase G0 or cell quiescence).

In contrast, in the presence of CycD, all trajectories lead to a unique dynamical cycle, made of a sequence of seven successive states (Figure 2, bottom left). From a qualitative point of view, the order of activity switching (off or on) matches the available data, as well as the time plots published by Novak and Tyson (2004).

Looking more carefully at this synchronous cycle, one can note that only two arrows correspond to single transitions, namely the activations of CycA and Cdc20, whereas three arrows correspond to double transitions, and two arrows to triple ones. In such situations, the synchronous approach impedes any further refined analysis of these transitions.

One may also consider a fully asynchronous assumption and generate all the trajectories compatible with the regulatory graph and the logical rules. Naturally, the stable state is conserved and can still be reached from all states lacking CycD activity.

Similarly, in the presence of CycD activity, the system has a unique attractor, but this now includes many intertwined cycles (see Figure 2, top; see also the web supplementary material for higher resolution graphs). Composed by 112 states, this attractor is a terminal strongly connected component in the sense of graph theory. In addition, the part of the state transition graph with CycD active encompasses several dozens of additional (non terminal) strongly connected components, each involving a small number of states (typically four) and potentially representing transient oscillations of few components on the way to the canonical cell cycle.

Interestingly, the seven states forming the synchronous cycle are also found in the terminal strongly connected component found in the asynchronous transition graph (see grey shaded states in Figure 2, top), together with the corresponding single transitions. However, the synchronous transitions may now correspond to multiple asynchronous paths.

2.3 Mixed a/synchronous updating

Between these two extreme updating assumptions, it is possible to define middle terms. One option is to consider the possibility that the realisation of some transitions requires several updating steps. Chaves *et al*. (2005) have recently explored this option to improve their Boolean model analysis of the segment polarity network

Table 1. Logical rules underlying the definition of the logical parameters associated with the regulatory graph of Figure 1

Product	Logical rules leading to an activity of the product	Justification/References
CycD	*CycD*	CycD is an input, considered as constant.
Rb	$(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{CycD} \wedge \overline{CycB})$	Rb is expressed in the absence of the cyclins, which inhibit it by phosphorylation (Novak and Tyson, 2004; Taya, 1997); it can be expressed in the presence of CycE or CycA if their inhibitory activity is blocked by p27 (Coqueret, 2003).
E2F	$(\overline{Rb} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{Rb} \wedge \overline{CycB})$	E2F is active in the absence of Rb, that blocks E2F self-transcriptional activation (Helin, 1998), and in the absence of CycA and CycB, that inhibit E2F (Novak and Tyson, 2004); CycA may be present, if its inhibitory activity is blocked by p27 (Coqueret, 2003).
CycE	$(E2F \wedge \overline{Rb})$	CycE activity requires the presence of E2f and the absence of Rb (Helin, 1998).
CycA	$(E2F \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge \overline{(Cdh1 \wedge Ubc)}) \vee (CycA \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge \overline{(Cdh1 \wedge Ubc)})$	The transcription of CycA is activated by E2F in the absence of Rb, which blocks this activation (Helin, 1998), in the absence of Cdc20, as well as of the pair formed by Cdh1 and UbcH10, which both lead to the degradation of CycA (Harper *et al*., 2002; Rape and Kirschner, 2004); CycA is stable in the absence of its inhibitors Rb, Cdc20, and of the pair Cdh1 and UbcH10.
p27	$(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{(\boldsymbol{CycE \wedge CycA})} \wedge \overline{\boldsymbol{CycB}} \wedge \overline{\boldsymbol{CycD}})$	p27 is active in the absence of the cyclins; when p27 is already present, it blocks the action of CycE or CycA (but not both of them) by sequestration (Coqueret, 2003).
Cdc20	*CycB*	CycB indirectly activates Cdc20 (Harper *et al*., 2002).
Cdh1	$(\overline{CycA1} \wedge \overline{CycB}) \vee (Cdc20) \vee (p27 \wedge \overline{CycB})$	The activity of Cdh1 requires the absence of CycB and CycA, which inhibit it by phosphorylation (Harper *et al*., 2002); Cdc20 further activates Cdh1. (Novak and Tyson, 2004); p27 allows the presence of CycA, by blocking its activity.
UbcH10	$(\overline{Cdh1}) \vee (Cdh1 \wedge Ubc \wedge (Cdc20 \vee CycA \vee CycB))$	UbcH10 is active in the absence of Cdh1; this UbcH10 activity can be maintained in the presence of Cdh1 when at least one of its other targets is present (CycA, Cdc20, or CycB) (Rape and Kirschner, 2004).
CycB	$(\overline{Cdc20} \wedge \overline{Cdh1})$	CycB is active in the absence of both Cdc20 and Cdh1, which target CycB for destruction (Harper *et al*., 2002).

The names of the components of the regulatory graph of Figure 1 are listed in the first column. For each one, the second column gives the logical rules specifying its behaviour. More precisely, we have described only the situations where the component is activated (value of the corresponding Boolean variable set to 1), all other situations leading to an inactivation. This description is based on the classical logical formulation, where ''$\wedge$'' stands for ''AND'', ''$\vee$'' stands for (inclusive) ''OR'', and the negation is written by a bar over the term. As an example, considering the case of CycE, there are eight non-zero parameters attached to CycE, specifying the different combinations of incoming interactions which lead to an activation of CycE (cf. the GINML file on the *GINsim* website). These can be summarised by the logical formula ''E2F active and Rb not active, whatever the state of the other components''. Finally the last column provides some justifications for the logical rules, together with references.

involved in the segmentation of the trunk of Drosophila embryos. Here, we propose an alternative approach enabling the combination of synchronous and asynchronous assumptions depending on the regulatory element or on the nature of transition considered. Indeed, depending on available knowledge or on the biological questions addressed, it may be necessary to go into fine grain dynamical analysis for only a subset of regulatory components. To deal with these issues, the last version of *GINsim* enables the user to group components into different classes, and to assign a *priority level* to each of these classes. In case of concurrent transition calls, *GINsim* first updates the gene(s) belonging to the class with the highest *ranking*. For each regulatory component class, the user can further specify the desired updating assumption, which then determines the treatment of concurrent transition calls inside that class. When several classes have the same ranking, concurrent transitions are treated under an asynchronous assumption (no priority).

To illustrate this approach, we have first built two priority classes, which arguably group faster *versus* slower biochemical processes. In the highest ranked transition priority class, we have included the degradations of E2F, CycE, CycA, Cdc20, UbcH10, CycB, as well as all transitions (in both directions) for CycD, Rb, p27 (Kip1) and Cdh1. The remaining transitions corresponding to synthesis rates (of E2F, CycE, CycA, Cdc20, UbcH10, and CycB) are grouped in a lower priority class. Using these two priority classes, both considered under the asynchronous assumption, we still obtain a single terminal strongly connected component (not shown) involving 34 states (to compare with the seven states obtained with the standard synchronous treatment, *versus* the 112 states in the fully asynchronous case without priority).

The analysis of this component reveals that some pathways are clearly unrealistic, as they skip the activation of some crucial cyclins, for example. To eliminate these spurious pathways, one can further refine the priority classes, taking into account additional information. Here, we can exploit the fact that several transitions are controlled by similar regulatory mechanisms and group them into synchronous classes. This leads to the definition of the four transition classes displayed in Table 2.

For this last prioritisation, we obtain a smaller terminal strongly connected component involving 18 states, which combine single and multiple transitions. This mixed graph is thus much simpler that the fully asynchronous transition graph. This graph enables a finer description of the sequence of events characteristic of the normal cell cycle than in the fully synchronous case. However, the data presently available do not allow a clear distinction between the different alternative pathways.

2.4 Mutant simulations

Beyond a faithful reproduction of the wild-type behaviour, a good cell cycle model should enable the simulation of various types of

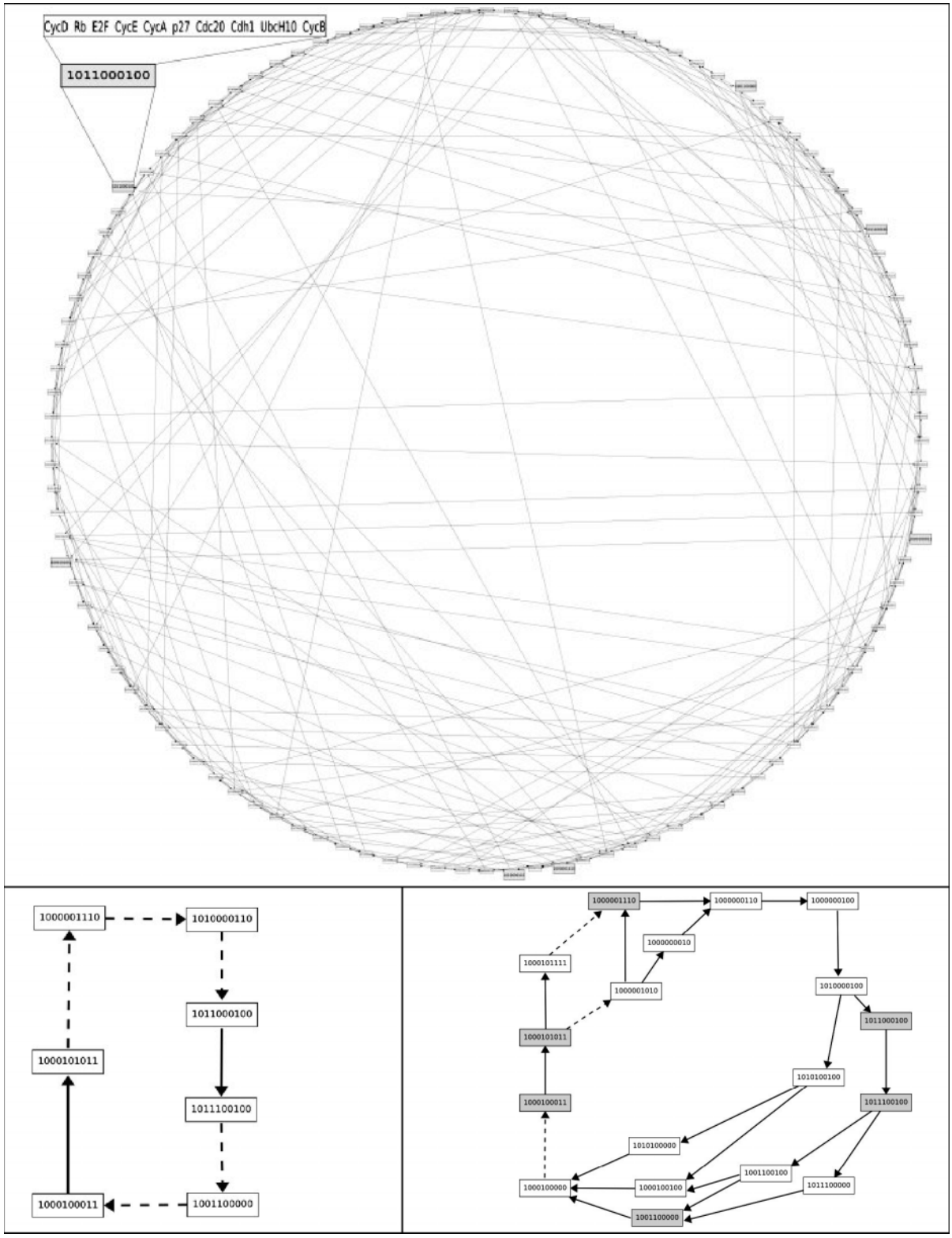


Fig. 2. Simulations of the wild-type cell cycle based on the Boolean model defined in Figure 1 and Table 1. Each vertex (node) represents one state, with the regulatory components ordered as mentioned in the top panel. The three state transition graphs correspond to the comprehensive asynchronous (top), the synchronous (bottom left), and a mixed (bottom right) assumptions. Note the difference of complexity between the asynchronous and synchronous graphs. In the bottom panels, solid arrows stand for single transitions, and dotted arrows for multiple transitions. The seven states involved in the synchronous cycle are grey-shaded in the asynchronous and mixed state transition graphs. For larger resolution pictures, see *GINsim* website.

Table 2. Priority transition classes used to obtain the strongly connected component shown in the bottom right panel of Figure 2

Rank	Type	Transitions
1	Asynchronous	CycD, Rb, p27, Cdh1, E2F↓, CycE↓
1	Synchronous	CycA↓, Cdc20↓, Ubc↓, CycB↓
2	Asynchronous	E2F↑, CycE↑, CycA↑, Cdc20↑
2	Synchronous	Ubc↑, CycB↑

The symbols ↓ and ↑ specify the (decreasing or increasing) direction of the considered transitions (by default, both directions are considered, *e.g.*, for CycD).

perturbations, in particular, the addition of drugs interfering with cell growth or cyclin activities, or the presence of loss-of-function or gain-of-function mutations in some of the core regulatory genes.

In this respect, *GINsim* provides a simple interface to constraint selected regulatory components within specific value intervals. Depending on the initial state(s), once a regulatory component has reached the corresponding value interval in the course of the simulation, all transitions leading outside of this interval are automatically discarded. This function greatly eases the simulation of loss-of-function and gain-of-function mutants (a table compiling our mutant analyses is maintained on the GINsim web page). As we have represented molecule families by single components, the comparison between *in silico* and experimental results are not always straightforward. However, up to now, all our simulation results are consistent with available experimental data (loss *versus* preservation of cell cycle depending on the mutant considered), but a few exceptions like the case of the p27 loss-of-function, for which our model predicts a stable state in the absence of CycD, whereas published data support the existence of oscillations in this situation. This discrepancy is likely due to the crude representation of CycE and Rb activity levels in terms of Boolean variables and could be solved by using ternary variables for these elements.

3 CONCLUSIONS AND PROSPECTS

In this paper, we have assessed the power of the logical approach, already in its simplest Boolean form, for the modelling of a complex protein interaction network. We have further presented extensions of our software *GINsim* to enable detailed studies of the asymptotical behaviour of complex systems, with synchronous, asynchronous or mixed treatment of concurrent transitions.

As shown through the analysis of a model for the mammalian cell cycle, a relatively simple logical model captures most qualitative dynamical features of the wild-type network, as well as of documented mutants. Strikingly, even simplistic synchronous simulations give rise to (only) two attractors consistent with available data, as well as with the simulations of Novak and Tyson (2004). On the one hand, we obtain a stable state in the absence of CycD, which matches our knowledge of quiescent cellular states when growth factors are lacking. On the other hand, in the presence of CycD, all trajectories converge towards a unique complex dynamical cycle. This is a favourable situation for the synchronous assumption, as no spurious cycle is generated.

However, the synchronous dynamics obtained does not allow the temporal separation of multiple regulatory activity changes. In contrast, asynchronous updating does allow finer temporal analyses, but the resulting state transition graph is very complex and encompasses many incompatible or unrealistic pathways. Leaning on specific *GINsim* functions, we have thus considered systematic ways to combine synchronous and asynchronous transitions, taking advantage of existing information on kinetics or regulatory mechanisms. This application thus illustrates the flexibility of the combination of different updating assumptions.

The logical formalism used should further enable the identification of the regulatory circuits playing the most crucial dynamical roles (Thomas *et al*., 1995). Our present model comprises 132 different circuits, involving from one to nine regulatory elements. A preliminary analysis suggests that only a dozen of these circuits are functional in some region of the variable space, most of the time only in the absence of CycD. The precise role of these different circuits has still to be clarified.

Modelling the molecular regulatory network controlling mammalian cell cycle is clearly a challenging and long-term enterprise. Focusing on the core network controlling the mammalian cell cycle, our present Boolean model corresponds to a relatively high level abstraction of our knowledge of the cellular system, which involves many variants for several of the molecular species considered (E2F, RB. . .). In this respect, the generation of extensive functional genomics data sets should prove of great help to delineate the specific expression and interaction patterns of these variants (for a pioneering attempt to exploit various kinds of functional genomic data sets to dynamically characterise the molecular network controlling the cell cycle in yeast, see the recent article by de Lichtenberg *et al*., 2005). On the basis of our generic, abstract model, several extensions or refinements can now be considered, including the use of multilevel variables wherever biological justifications can be advanced, further specifications and enrichments of this model in reference to specific cell types, or yet the inclusion of additional control modules.

A substantial increase in the sophistication of the logical models considered will lead to combinatorial problems, *e.g.*, to identify all attractors or to analyze the trajectories leading to these attractors. To prepare the ground to deal with such combinatorial problems, we are exploring different approaches. First, we use constraint programming to delineate attractors from simple or composed logical models without computing the whole state transition graph (Devloo *et al*., 2003). Next, we have developed and implemented a set of translations rules enabling the export of parameterised regulatory graphs into standard or coloured Petri nets, thereby enabling the use of the various dynamical analysis tools developed by this lively community (Chaouiya *et al*., in press). Finally, we are presently evaluating the application of temporal logic formalisms (*e.g.*, Computational Tree Logics) to assess the existence of specific dynamical pathways, or to encompass specific temporal information (Bernot *et al*., 2004; Batt *et al*., 2005).

Ultimately, more quantitative models are needed to explore fine grain aspects of the control of the cell cycle, *e.g.*, modulations of the cycle period or of its amplitude. In this respect, Petri nets constitute an interesting framework to refine discrete models, leaning on existing hybrid or stochastic extensions. Alternatively, one may use sets of differential or stochastic equations, but even in this case, a preparatory logical analysis should prove useful when dealing with large and complex regulatory networks.

ACKNOWLEDGEMENTS

We wish to thank A. Ciliberto and K. Helin, as well as E. Remy, P. Ruet and B. Mossé, for insightful discussions on biological, and mathematical aspects of this work. We acknowledge financial support from the European Commission (contract LSHG-CT-2004-512143), the French Research Ministry (ACI IMPbio), the CNRS, and the INRIA (ARC MOCA).

REFERENCES

Batt,G., Ropers,D., de Jong,H., Geiselmann,J., Mateescu,R., Page,M., Schneider,D. (2005) Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in *Escherichia coli*. *Bioinformatics*, **21**, i19–i28.

Bernot,G., Comet,J.P., Richard,A., Guespin,J. (2004) Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. *J. Theor. Biol*., **229**, 339–347.

Chaouiya,C., Remy,E., Mossé,B., Thieffry,D. (2003) Qualitative analysis of regulatory graphs: A computational tool based on a discrete formal framework. *Lect. Notes Control Inf. Sci*., **294**, 119–126.

Chaouiya,C., Remy,E., Thieffry,D. (in press) Petri net modelling of biological regulatory networks. *J. Discrete Algorithms*.

Chaves,M., Albert,R., Sontag,E.D. (2005) Robustness and fragility of Boolean models for genetic regulatory networks. *J. Theor. Biol*., **235**, 431–449.

Coqueret,O. (2003) New roles for p21 and p27 cell-cycle inhibitors: a function for each cell compartment? *Trends Cell Biol*., **13**, 65–70.

de Jong,H. (2002) Modelling and simulation of genetic regulatory systems: A literature review. *J. Comput. Biol*., **9**, 67–103.

de Lichtenberg,U., Jensen,L.J., Brunak,S., Bork,P. (2005) Dynamic complex formation during the yeast cell cycle. *Science*, **307**, 724–727.

Devloo,V., Hansen,P., Labbe,M. (2003) Identification of all steady states in large networks by logical analysis. *Bull. Math. Biol*., **65**, 1025–1051.

Dimova,D.K., Dyson,N.J. (2005) The E2F transcriptional network: old acquaintances with new faces. *Oncogene*, **24**, 2810–2826.

Fuß,H., Dubitzky,W., Downes,C.S., Kurth,M.J. (2005) Mathematical models of cell cycle regulation. *Brief. Bioinformatics*, **6**, 163–177.

Glass,L., Kauffman,S.A. (1973) The logical analysis of continuous non-linear biochemical control networks. *J. Theor. Biol*., **39**, 103–129.

Gonzalez,A.G., Naldi A.,Sánchez,L., Thieffry,D., Chaouiya,C. (2006) GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. *Biosystems*, **84**, 91–100.

Harper,J.W., Burton,J.L., Solomon,M.J. (2002) The anaphase-promoting complex: it's not just for mitosis anymore. *Genes Dev*., **16**, 2179–2206.

Helin,K. (1998). Regulation of cell proliferation by the E2F transcription factors. *Curr. Opin. Genet. Dev*., **8**, 28–35.

Huang,J.N., Park,I., Ellingson,E., Littlepage,L.E., Pellman,D. (2001) Activity of the APCCdh1 form of the anaphase-promoting complex persists until S phase and prevents the premature expression of Cdc20p. *J. Cell Biol*., **154**, 85–94.

Kauffman,S.A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press.

Novak,B., Tyson,J.J. (2004) A model for restriction point control of the mammalian cell cycle. *J. Theor. Biol*., **230**, 563–579.

Olashaw,N., Bagui,T.K., Pledger,W.J. (2004) Cell Cycle Control A Complex Issue. *Cell Cycle*, **3**, 263–264.

Rape,M., Kirshner,W.W. (2004) Autonomous regulation of the anaphase-promoting complex couples mitosis to S-phase entry. *Nature*, **432**, 588–595.

Tamrakar,S., Rubin,E., Ludlow,J.W. (2000) Role of pRb dephosphorylation in cell cycle regulation. *Front. Biosci*., **5**, 121–137.

Taya,Y. (1997) RB kinases and RB-binding proteins: new points of view. *Trends Biochem. Sci*., **22**, 14–17.

Tessema,M., Lehmann,U., Kreipe,H. (2004) Cell cycle and no end. *Virchows Arch*., **444**, 313–323.

Thomas,R. (1991) Regulatory networks seen as asynchronous automata: a logical description. *J. Theor. Biol*., **153**, 1–23.

Thomas,R., Thieffry,D., Kaufman,M. (1995) Dynamical behaviour of biological regulatory networks—I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bull. Math. Biol*., **57**,247–276.

Zhang,H.S., Gavin,M., Dahiya,A., Postigo,A.A., Ma,D., Luo,R.X., Harbour,J.W., Dean,D.C. (2000) Exit from G1 and S phase of the cell cycle is regulated by repressor complexes containing HDAC-Rb-hSWI/SNF and Rb-hSWI/SNF. *Cell*, **101**, 79–89.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e132–e140
doi:10.1093/bioinformatics/btl219

Semi-supervised LC/MS alignment for differential proteomics

Bernd Fischer[1,*], Jonas Grossmann[2], Volker Roth[1], Wilhelm Gruissem[2], Sacha Baginsky[2] and Joachim M. Buhmann[1]

[1]Institute of Computational Science, ETH Zurich and [2]Institute of Plant Sciences, ETH Zurich, Switzerland

ABSTRACT

Motivation: Mass spectrometry (MS) combined with high-performance liquid chromatography (LC) has received considerable attention for high-throughput analysis of proteomes. Isotopic labeling techniques such as ICAT [5,6] have been successfully applied to derive differential quantitative information for two protein samples, however at the price of significantly increased complexity of the experimental setup. To overcome these limitations, we consider a *label-free* setting where correspondences between elements of two samples have to be established prior to the comparative analysis. The alignment between samples is achieved by nonlinear robust ridge regression. The correspondence estimates are guided in a semi-supervised fashion by prior information which is derived from sequenced tandem mass spectra.

Results: The semi-supervised method for finding correspondences was successfully applied to aligning highly complex protein samples, even if they exhibit large variations due to different biological conditions. A large-scale experiment clearly demonstrates that the proposed method bridges the gap between statistical data analysis and label-free quantitative differential proteomics.

Availability: The software will be available on the website http://people.inf.ethz.ch/befische/proteomics

Contact: bernd.fischer@inf.ethz.ch

1 INTRODUCTION AND RELATED WORK

A widely used approach to the sample-alignment problem fits a piece-wise linear function to maximize the correlation between the two samples. Methods of this kind are often characterized as *correlation optimized warping* (COW) (12). Other approaches are based on *hidden Markov models* (HMM) which formally define generative models for aligned samples, see e.g. Listgarten *et al*., (11). From a machine learning perspective, both COW and HMM methods are purely *unsupervised* in nature, since they do not exploit prior information of known correspondences. Both approaches share also the commonality that they have been solely applied to aligning *total ion counts*. Figure 1 depicts total ion count curves for two samples under two different biological conditions. Aligning these two samples is very difficult when the total ion counts are exclusively used as the information source.

In principle, both COW and HMM can be extended to aligning multi-dimensional data. It is, however, extremely difficult to handle LC/MS data of complex samples which are typically characterized by a very large input dimension (up to a mass range of 2500 Da for doubly charged peptides). The data analysis situation becomes even more complicated if we have to align highly heterogeneous samples that were taken under different biological conditions. Under these conditions one typically finds many peaks that do not match to *any* other peak in the second sample.

A first attempt to overcome these problems was made by Tibshirani *et al*. (14), who introduced an aligning technique based on hierarchical clustering.

In this paper we describe a new approach for LC/MS alignment exploiting additional information from sequenced tandem mass spectra rather than aligning only peaks from the LC/MS image. The second spectrometry stage is used to acquire sequence information. From a subset of these sequences which are identified in *both* samples, a time warping function is estimated by fitting a nonlinear regression function. Since there exists a number of false-identifications we use a *robust* regression model to reduce the sensitivity to outliers. Starting from an initial alignment hypothesis, we further improve the model by combining supervision information (sequenced peaks) and unlabeled information (all other peaks) within an iterative *self-training* scheme: the predictive variance is computed for each of the peaks, and peaks with a very small uncertainty are assigned a target value. Then, the model is re-trained based on the enlarged dataset, and the whole procedure is iterated until all peaks are labeled. This inclusion of unlabeled data yields an improved detection of peak correspondences. All free model parameters are selected by employing a cross-validation loop. With this novel machine learning technique we are able to align the underlying experiments of Figure 1.

2 EXPERIMENTAL SETTING AND DATA GENERATION

2.1 Liquid chromatography and mass spectrometry

Before analyzing the proteins in a cell, the proteins are digested by a specific enzyme like Trypsin, resulting in a mixture of small peptides. The peptides are separated by high-performance liquid chromatography. At (almost) equally spaced retention time steps a mass spectrum is acquired from the peptide sample eluting from the LC-column. The recording of a mass spectrum requires that a peptide is ionized and transferred into the gas phase, typically by electro-spray ionization. Most of the peptides are doubly or triply charged, but singly charged peptides also appear in proteomics experiments. The data are represented in form of a two dimensional

*To whom correspondence should be addressed.

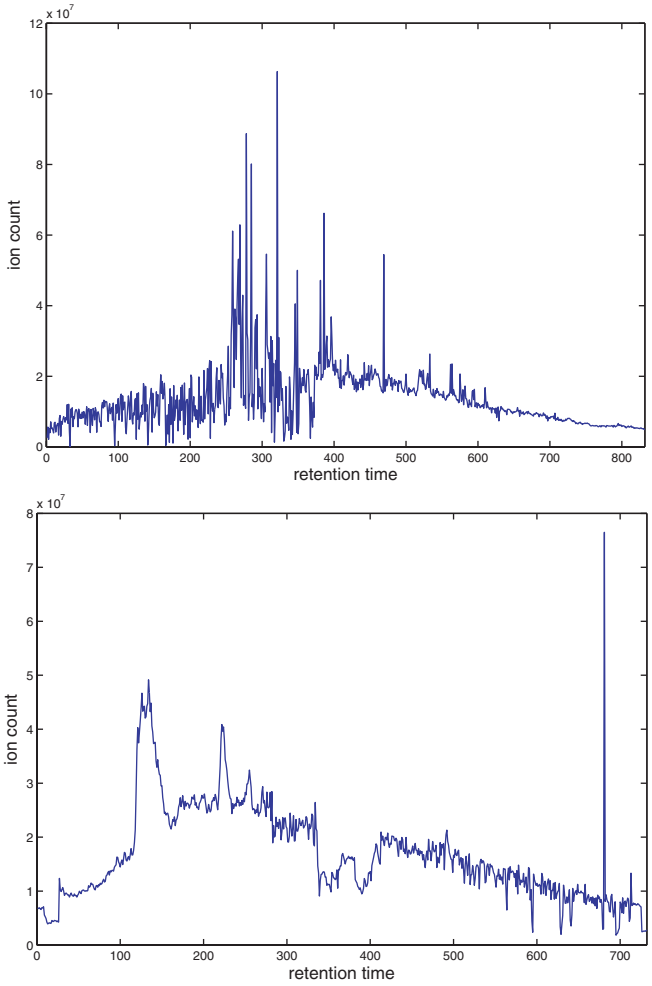


Fig. 1. The total ion count per time unit of two protein samples under two different biological conditions.

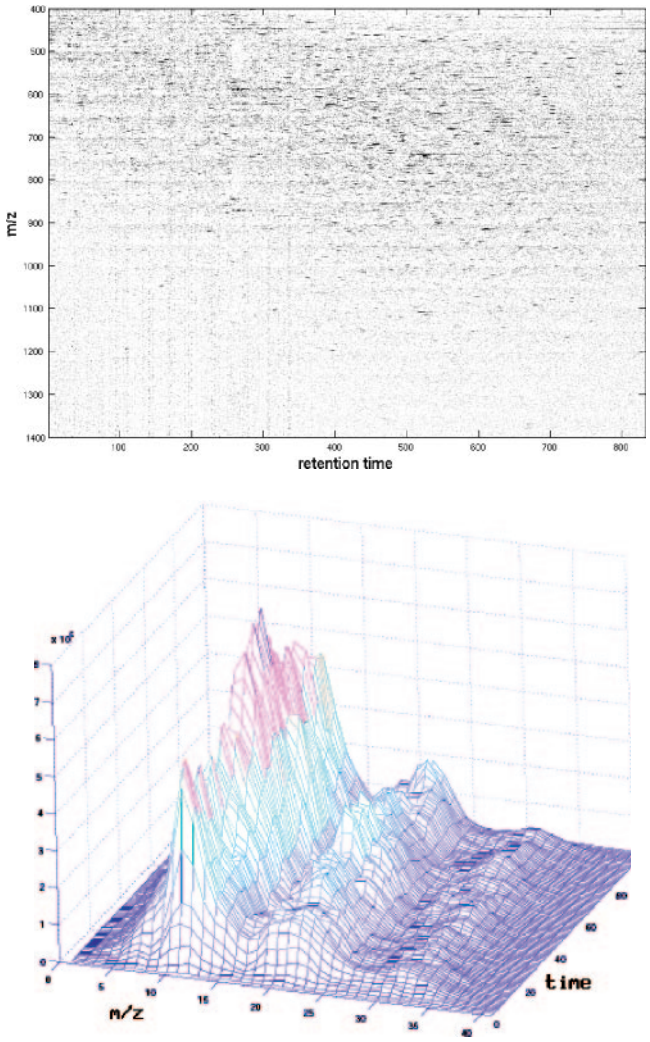


Fig. 2. Top: LC/MS image. The x-axis is the retention time, the y-axis is the peptide mass. Bottom: One peak in the LC/MS image accumulated over all singly charged peptides.

measurement, where one dimension is the retention time (t) and the other dimension is defined by the peptide mass over charge (m/z) (See Figure 2). We will refer to this two dimensional measurement as the LC/MS image. The local maxima in the LC/MS image correspond to different peptides with different m/z values. The bottom figure shows an accumulation of peaks over a large number of singly charged peptides. One can recognize three different isotopes for each peptide. Isotopes are common, since peptides are composed of a large amount of C-atoms. The integral over the peak area m_i yields the amount of ions of a specific peptide i.

2.2 Quantitative measurement

The over-all goal of quantitative proteomics is the estimation of the absolute protein expression. Let $I(p)$ denote the set of peptide indices for protein p. Assuming that all peptides $I(p)$ of a protein produce the same amount of ions and assuming a log-normal error distribution, one can estimate the log protein expression as

$$\widehat{\log e}_p = \frac{1}{|I(p)|} \sum_{i \in I(p)} \log m_i \,. \qquad (1)$$

The log-normal error model seems to describe expression levels well in practice, although we are not aware of any systematic study of this observation. The other assumption, however, that all peptides produce the same amount of ions rarely holds. The peak-area integrals are typically quite different for peptides of the same protein. One reason lies in the ionization efficiency of the peptides and suppression effects between peptides. An incomplete or overcomplete digestion process can also contribute to this discrepancy. There is still too little known about the reason for the different behavior of peptides. These uncertainties in the measurements render absolute quantitative proteomics infeasible today, but for peptide specific multiplicative errors, the ratio of peak area integrals can reliably be estimated. In our experience this assumption holds as long as the two samples are fairly similar. For two very different samples, peptide unspecific suppression effects play the major role.

Given two samples that both contain a certain peptide i and two corresponding measurements $m_i^{(1)}$ and $m_i^{(2)}$, the log-protein ratio in

both samples can be estimated by

$$\log r_p = \frac{1}{|I(p)|} \sum_{i \in I(p)} \log \frac{m_i^{(1)}}{m_i^{(2)}} . \quad (2)$$

A common procedure to measure peptides under two conditions is isotopic labeling like ICAT (5,6). The peptides in the two samples are marked with labels of different weights. The two samples are then mixed together and measured together. In the resulting LC/MS image, peptides of the two samples occur with a mass shift corresponding to the different weights of the labels. In addition to the fact that labels are still expensive, this approach carries the disadvantage that the two samples have to be mixed together. In many applications, however, it is advantageous to measure both samples separately. For example in cancer detection, one would like to first analyze a certain number of collected disease samples which then can be compared with patient probes without analyzing the disease samples over and over again.

Label-free techniques do not suffer from these shortcomings. Without the label information, on the other hand, one is forced to detect corresponding peaks in the two samples. In order to solve this correspondence problem we first have to shed some light on the procedure of *peak picking* which extracts peaks in the LC/MS image.

2.3 Peak detection

At the beginning of the analysis process, the mass spectrometry data is stored in a large data matrix, the columns of which represent mass spectra taken at different retention times. The *m/z* axis of these spectra is discretized in 1.00045 *Da* bins which can be justified as follows: If an amino acid is divided by its elementary mass (the number of protons and neutrons), the average mass of one elementary unit (a proton or neutron) is 1.00045 *Da*. Thus a peptide with 2000 elementary units has a mean mass of 2000.9 *Da*. The difference of 0.9 *Da* to the naively expected mean mass of 2000 *Da* is clearly detectable by our mass spectrometer and this mass correction significantly increases e.g. the peptide retrieval in de novo sequencing (3).

To ensure a standardized representation, each mass spectrum is normalized by its total ion count, i.e. by the sum over the spectrum. In the next step of the analysis process we measure the background noise level by median filtering over a window of ±50 in time and mass direction. This estimated noise level is then subtracted from the measurements. An entry in the LC/MS matrix is marked as a *peak area*, if the mean over ±5 in time direction and +1 in mass direction exceeds at least 3.0 times the mean over pixels surrounding the potential peak. The local maximum in each connected component defines the peak position with time and mass coordinates. Figure 3 shows the detected peaks in the LC/MS image.

2.4 Sequence identifications

At this stage of the analysis process the amino-acid sequence of the detected peaks is not available. We can, however, acquire sequence information for a certain fraction of peaks by way of *Tandem mass spectrometry*. From a measured MS spectrum a MS/MS device selects one of the peaks exceeding a predefined level. The ions in a small mass window around the selected mass are stabilized in an ion trap and fragmented by collision with a noble gas. The mass spectrum of the fragment ions contains information about the peptide sequence. The tandem mass spectra are denoted MS/MS spectra to distinguish them from standard MS spectra. Searching the spectrum against a database (2,7) produces hypotheses about the underlying peptide sequence. The hypothesized sequences are then validated by using *PeptideProphet* (10). In our experiments we consider spectrum identifications with a posterior probability $p \geq 0.97$ as being valid. Successful sequence identification without database knowledge is still a challenging problem. We have shown that small subsequences can be identified by *de novo* peptide sequencing (3) in many cases. In this work, however, we only use the database search results.

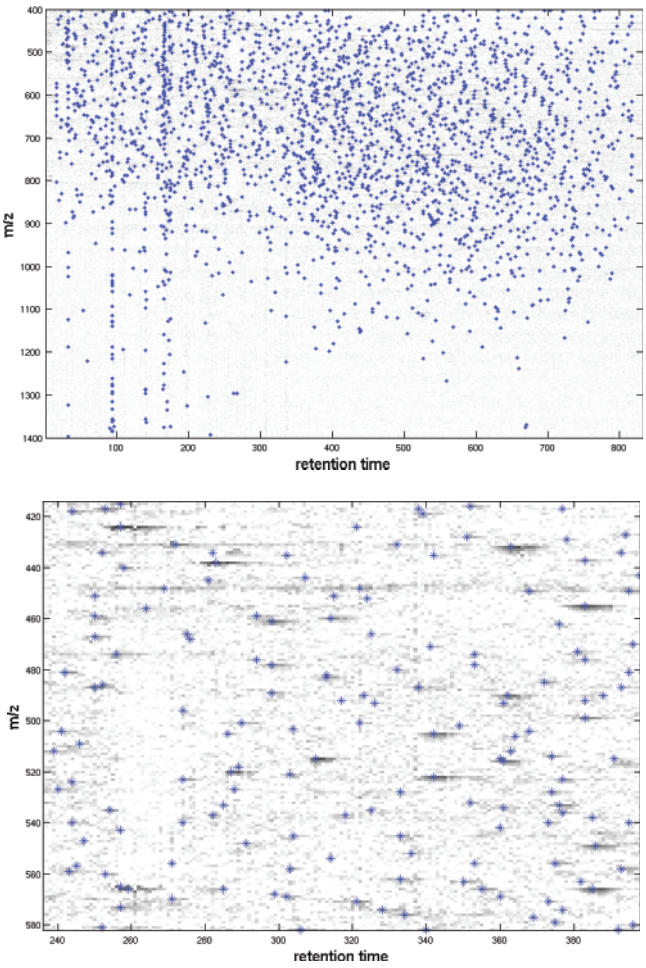


Fig. 3. Top: The detected peaks in the LC/MS image. Bottom: detailed view of sub-image.

To identify each MS/MS spectrum with one of the detected peaks in the LC/MS image, we search for a detected peak in the neighborhood of the mass/time coordinate of the MS/MS spectrum. We observed that in most cases the mass of the detected peak is correct or increased by 1 Da. Such increments might occur, if the first isotope is much larger than the mono-isotopic peak. Figure 4 depicts the fraction of sequenced MS/MS spectra that can be assigned to a peak. The quantity $w0$ denotes the size of the window, in which a peak is accepted if the mass is correct, and $w1$ is the corresponding window for mass differences of one. The asymmetry in the figure shows that the majority of peaks have the correct mass. Choosing

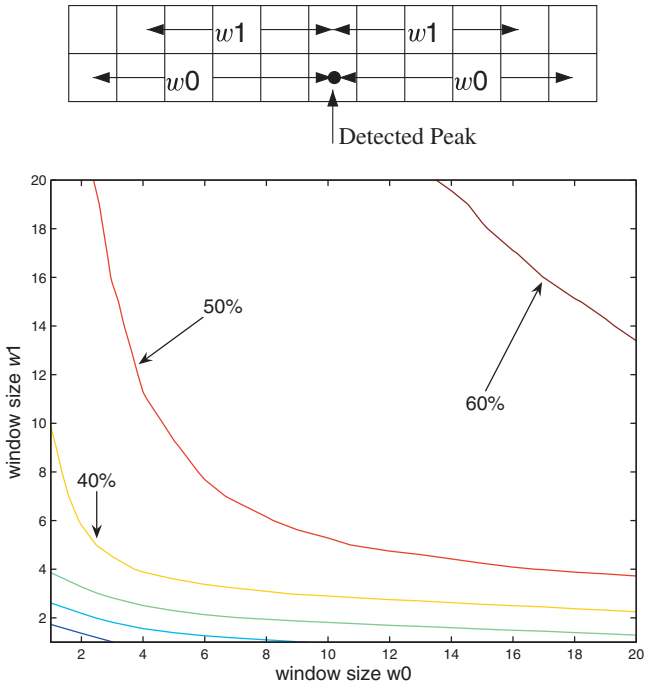


Fig. 4. Top: The window in which an identified spectrum is assigned to a peak. Bottom: The fraction of identified spectra that can be assigned to a peak for varying window sizes. Contour lines are spaced in 10% intervals.

$w0 = 10$ and $w1 = 5$ we can assign 52.2% of all identified sequences to a peak. This rate might be increased by using larger windows, however at the price of a higher false-positive rate. We will discuss this issue in more detail later. The large fraction of not assignable sequenced spectra is due to peaks that can hardly be distinguished from the background.

The search includes all singly, doubly, and triply charged peptides. Denoting the mono-isotopic mass of a peptide by m, the m/z-values of a singly ($i = 1$), doubly ($i = 2$) and triply ($i = 3$) charged peptide are observed as mass/charge ratios

$$\frac{m^{(i)}}{z} = \frac{m+i}{i} \tag{3}$$

due to protone capture. On average there are about 5000 peaks per LC/MS image from which roughly 200 could be sequenced. Figure 5 depicts the distribution of the different charge states over the LC/MS image. The green circles show the singly charged peptides, the red crosses are the doubly charged peptides and the blue filled circles are the triply charged ones.

2.5 Scenarios in quantitative proteomics

The analysis process described above extracts two different types of information from the mass spectrometry data:

- a list of peaks in the LC/MS image, and
- sequence information for a small subset of the peak list.

A quantitative analysis based on these input data can pursue different goals:

(i) in a **classification scenario** one would like to separate a certain protein sample under one biological condition (extracted e.g. from a diseased patient) from samples under another biological condition (extracted e.g. from a control group). For the mere classification task one does not need the peptide sequence information. One rather tries to find as many corresponding peaks between two samples of different biological conditions as possible.

(ii) A different scenario is known as **biomarker discovery** (9). In addition to classification, one would like to identify proteins or peptides which are causally related to a certain biological condition (e.g. a certain disease). From a machine learning point of view this identification problem defines a *feature selection* task. Having selected 'relevant' features one is typically interested in the underlying sequences. Thus, if we pursue biomarker discovery as our goal, we have to *sequence* as many peptides as possible. Ultimately, we try to compare the complete proteome using these processing steps.

In this paper we will show that the number of peak correspondences (for classification) as well as the number of sequence identified correspondences (for biomarker discovery) can be increased by combining labeled and unlabeled information.

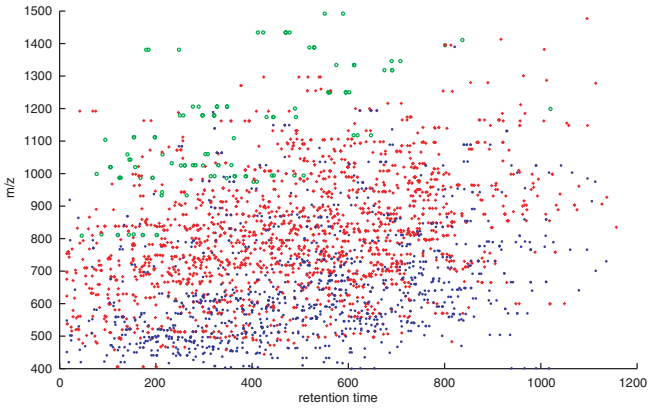


Fig. 5. Distribution of charge values over LC/MS image. Green circles: singly charged peptides. Red crosses: Doubly charged. Blue filled circles: Triply charged.

2.6 Sample preparation

The peptides we used for the analysis were derived from plant cell culture samples that were exposed to different illumination programs (light versus dark). The proteins are fractionated by SDS-PAGE and in-gel digested. The peptide mixture was loaded onto a C18 reversed phase column and eluted with a gradient developed from solvent A (5% ACN, 0.2% formic acid) and solvent B (80% ACN, 0.2% formic acid). Gradient shape was as follows: 26 minutes 100% solvent A, within 0.2 minutes up to 5% solvent B, within additional 69 minutes up to 55% solvent B and in one additional minute up to 100% B. The flow rate at the tip of the column was adjusted to ≈200 nl/min. The chromatography (LC) was coupled online to an LTQ ion trap mass spectrometer (Thermo-Finnigan, San Jose, CA, USA) equipped with a nanospray ionization source. Mass analysis was performed with a spray voltage of 2.0–2.5 kV and one MS full scan followed by three data-dependent MS/MS scans of the three most intensive parent ions. The dynamic exclusion function was enabled to permit one measurement of a particular

parent ion followed by an exclusion of the aquisition of MS/MS spectra for this parent ion over a periode of 4 min.

3 LC/MS ALIGNMENT

When comparing two subsequent LC/MS scans, slight changes of the time scale can often be observed in different experimental situations. To compensate these time differences, an alignment function of the form $f : t^{(1)} \mapsto t^{(2)}$ maps the time scale of one experiment to that of the second experiment. Instead of directly mapping the scale itself, one can alternatively map one scale to the *scale differences* between the two samples:

$$g : t^{(1)} \mapsto t^{(2)} - t^{(1)}. \tag{4}$$

This formulation provides a clear visualization of the inherent non-linearities of the warping process. Within the subset of peaks sequenced in the second MS stage, we typically find an overlap of 10-70 identified peaks that are common in both experiments. Figure 6 depicts such time-warping functions learned from the subset of common peptides for two different pairs of biological samples. The non-linear relationship between the time-scales is clearly visible in the top panel.

3.1 Warping by way of robust regression

Identifying the $t_i^{(1)}$-values with x_i, and the time differences $t_i^{(2)} - t_i^{(1)}$ with y_i, the warping function depicted in Figure 6 is determined by first expanding the x-values in a k-th order polynomial basis

$$\phi_i := \phi(x_i) = (1, x_i, x_i^2, \ldots, x_i^k)^t, \tag{5}$$

and then by fitting a robust ridge-regression model. The latter finds the $k+1$ dimensional weight vector β which minimizes

$$\sum_{i=1}^{n} L_c(\phi_i^t \beta - y_i) + \lambda \beta^t \beta, \tag{6}$$

where $L_c(\xi)$denotes a robust loss function of Huber's type:

$$L_c(\xi) = \begin{cases} c\,|\,\xi\,| - \dfrac{c^2}{2}, & \text{for}\,|\,\xi\,| > c \\ \dfrac{\xi^2}{2}, & \text{for}\,|\,\xi\,| \le c. \end{cases} \tag{7}$$

Both the degree k of the polynomial and the ridge-penalty λ are chosen by 10-fold cross-validation. The reader should notice that the above nonlinear regression model is equivalent to using a *kernel regression model* with polynomial kernal of degree k. For computational reasons, in this special application it is better to *explicitly* expand the input data in the polynomial basis, rather than using the kernelized version.

In the usual regression setting, the observations y are assumed to be generated by corrupting the values of $f(x_i) = \phi_i^t \beta$ by additive noise that follows some density $p(\xi)$. Huber's loss function turns out to be optimal (in the sense that it guarantees the smallest loss in a worst case scenario), if the true noise density is a mixture of two components, one of which is known to be Gaussian distributed and the other one is an arbitrary density (8). Huber's loss function penalizes large deviations $|\,\xi\,| > c$ only linearly. Thus, it is superior to its standard quadratic counterpart in situations where the data contains outliers which are generated by an unknown and possibly

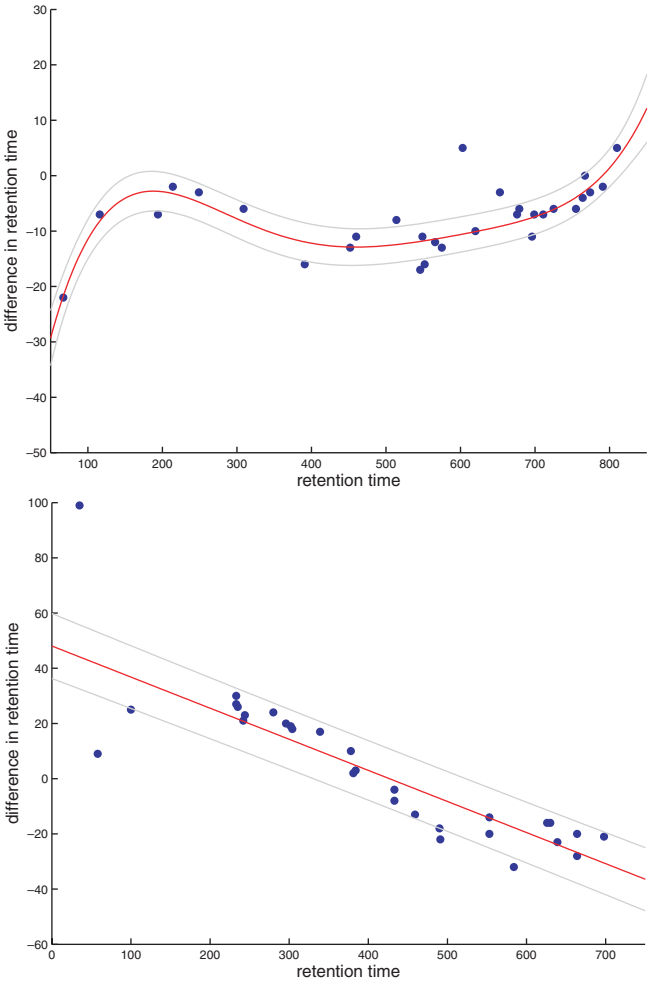


Fig. 6. Examples of two different alignments. On the x-axis the retention time is plotted, on the y-axis the difference in retention time. The red curve depicts the estimated warping function, the light gray ones show 1σ-confidence intervals.

highly fluctuating noise source. The parameter c is typically estimated from the data in an iterative fashion as a multiple of the standard deviation of the observed residuals. A common scaling formula is $c = 1.345\sigma$, which yields 95% efficiency when the errors are normal, and still protects against outliers. Usually a robust measure of spread is employed in preference to the standard deviation of the residuals. For example, a common approach is to choose $\hat{\sigma} = MAR/0.6745$, where MAR is the median absolute residual. This choice defines an unbiased estimator of the standard deviation for Gaussian data, see (4).

The optimal weight vector β that minimizes eq. (6) is found iteratively as the solution of a re-weighted least squares problem:

$$\beta^{\text{new}} = [\Phi^t \Omega(\beta) \Phi + 2\lambda I]^{-1} \Phi^t \Omega(\beta) y, \tag{8}$$

where Φ denotes the (transformed) data matrix with rows ϕ_i, and $\Omega(\beta)$ denotes the diagonal matrix

$$\Omega(\beta) = \text{diag}\{\omega([\Phi\beta - y]_i)\}, \tag{9}$$

with $\omega(\xi) := (1/\xi) \cdot \frac{\partial L_\varepsilon(\xi)}{\partial \xi}$. The final entries Ω_{ii} define weights for the individual training data $\phi(x)_i$.

3.2 Semi-supervised alignment

In the above derivation, the regression function is learned exclusively from the subset of identified correspondences in both samples. Due to technical limitations, the number of MS/MS spectra and thus the number of peptide sequence identifications is usually relatively small. We will now exploit the ideas of *self-training* (13), to additionally extract the information contained in the remaining peaks. Self-training is an incremental algorithm that labels the unlabeled data and converts the most confidently *predicted* data points into labeled training examples. This iteration proceeds until all the unlabeled data are consistently labeled. In order to apply this mechanism to our LC/MS alignment problem, we have to derive a formula for the *predictive uncertainty* of test data.

We denote by Φ_G the subset of training data which have been assigned a weight $\Omega_{ii} = 1$ in the robust regression procedure defined in eq. (8). These data points have small residuals $|\xi| \leq k$ which are penalized quadratically by the robust loss function eq. (7). Thus, for these points the *Gaussian* noise assumption is valid. Since in this case the posterior distribution is also Gaussian, a Bayesian treatment of regression allows us to derive an analytical expression for the uncertainty of the prediction for a new data point x_*:

$$\begin{aligned} Var[f(x_*)] &= E_{\beta|X}[(f(x_*) - E[f(x_*)])^2] \\ &= \sigma^2 \phi^t(x_*)(\lambda I + \Phi_G^t \Phi_G)^{-1} \phi(x_*). \end{aligned} \tag{10}$$

The total predictive variance, $Var[y(x_*)]$, is the sum of the noise variance σ^2 and the variance about the mean, $Var[f(x_*)]$, since both sources of variation are uncorrelated, see e.g. (1) for details. For estimating the noise variance one might again use the above equation $\hat{\sigma} = MAR/0.6745$ applied to the data in Φ_G.

Our adaption of the self-training method now proceeds as follows:

Initialize: train the model on the correspondences verified by sequencing.

Iterate:

(i) for a peak which elutes at time $t_i^{(1)}$ in the first LC/MS image, predict the time difference $t_i^{(2)} - t_i^{(1)}$;

(ii) for every such predicted peak, compute its predictive variance;

(iii) for the 10% most certain peaks, search for a corresponding peak in the second LC/MS image within a certain window.

(iv) include all found correspondences into the training set, and retrain the model;

Until: No more peaks are found within a 2σ-confidence interval around the current fit.

Figure 7 shows the outcome of this semi-supervised learning algorithm for the two samples that were analyzed previously in Figure 6. The labeled objects are colored dark blue. Compared to the alignment computed exclusively on the labeled objects (cf. Figure 6), the inclusion of unlabeled objects makes it possible to model more details of the warping function. Compared to the supervised solutions, where often only a straight line can be reliably fitted to the data, the semi-supervised solutions typically use regression models of higher complexity (measured in terms of the polynomial degree k in the expansion eq. (5), which is automatically selected by cross validation).

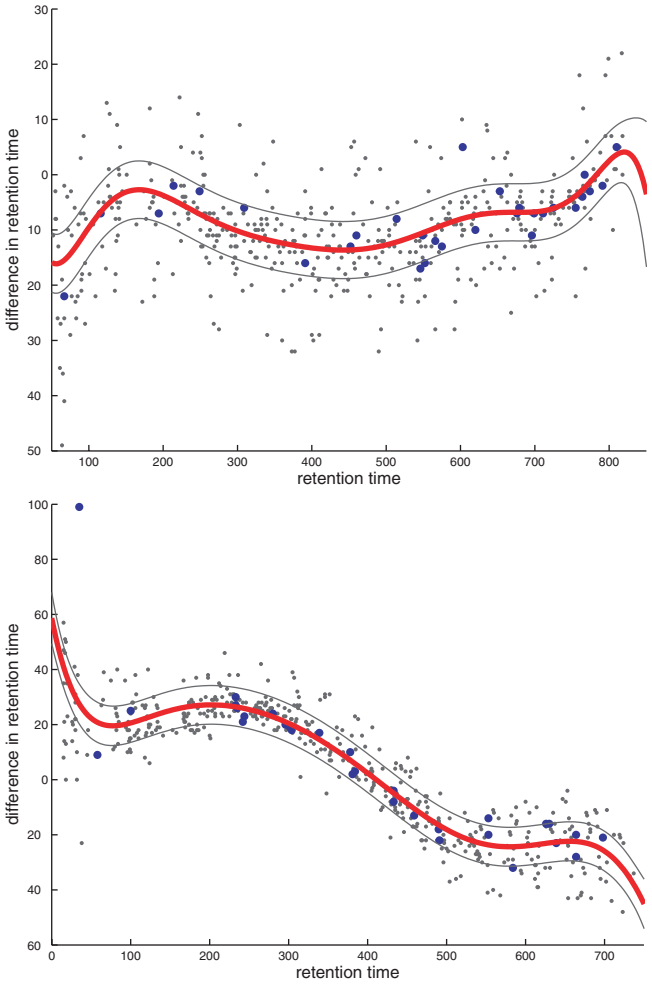


Fig. 7. Example of two different alignments for semi-supervised learning. On the x-axis the retention time is plotted, on the y-axis the difference in retention time. The blue (light gray) dots are the sequenced (non-sequenced) peaks. The light gray curves depict 1σ-intervals of the predictive uncertainty.

3.3 Detecting peak correspondences

First we analyze the performance of the alignment in the **classification** scenario, where all peaks (sequenced as well as unsequenced) are aligned. The alignment function computed by minimizing eq. (6) treats the two samples in an non-symmetrical fashion, since it warps the first time scale to the second. In order to derive symmetric correspondences between peaks, we predict the retention times in both directions separately, which allows us to easily check the self-consistency of the prediction model. Given a peak in sample A, our method predicts the retention time in sample B. If we have detected a peak in sample B within a window w around the predicted peak position, we denote this a (directed) correspondence. Here again we tolerate a mass difference of at most ±1 Da. Predicting retention time in both directions between sample A and sample B gives us a list of (directed) correspondences from sample A to sample B and a list of (directed) correspondences

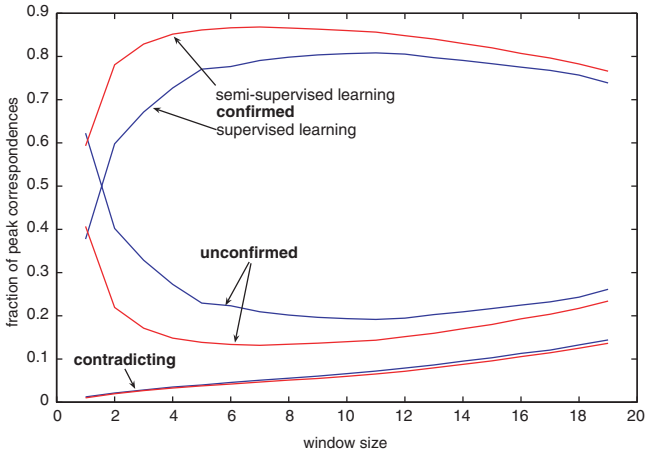


Fig. 8. Fractions of confirmed/unconfirmed/contradicting peak correspondences for the semi-supervised model (red) and the purely supervised model (blue) as a function of window size.

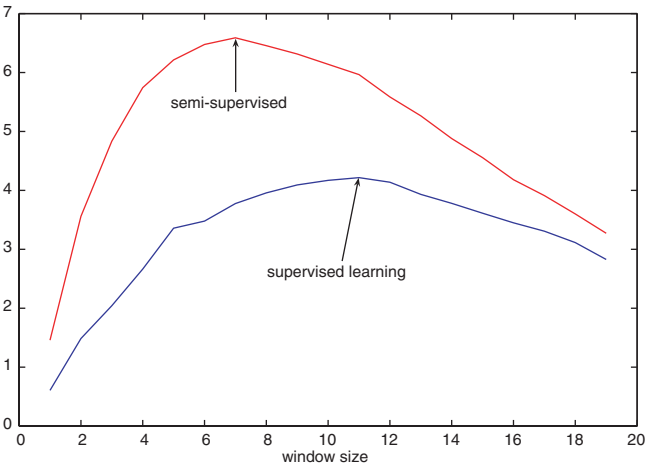


Fig. 9. Performance ratio #(confirmed)/[#(unconfirmed) + #(contradicting)] correspondences for the semi-supervised model (red) and the supervised model (blue).

from sample *B* to sample *A*. A correspondence is called *confirmed* if we find a correspondence in both directions. If we find a peak only in one of the directions, we call the correspondence *unconfirmed*. If we obtain two different mappings for one peak, we declare the 'correspondence' as *contradicting*. Denoting by n_1 the number of confirmed, by n_2 the number of unconfirmed and by n_3 the number of contradicting correspondences, the respective rates $n_i/(n_1 + n_2 + n_3)$ are depicted in Figure 8. It is obvious that the fraction of contradicting correspondences monotonically increases if the window is enlarged. For very small windows most correspondences remain unconfirmed, whereas the fraction of confirmed correspondences attains a maximum for windows of intermediate size. In practice, we have to balance the number of confirmed correspondences against the unconfirmed and/or contradicting ones. Figure 9 shows the quotient $n_1/(n_2 + n_3)$ both for the semi-supervised and supervised

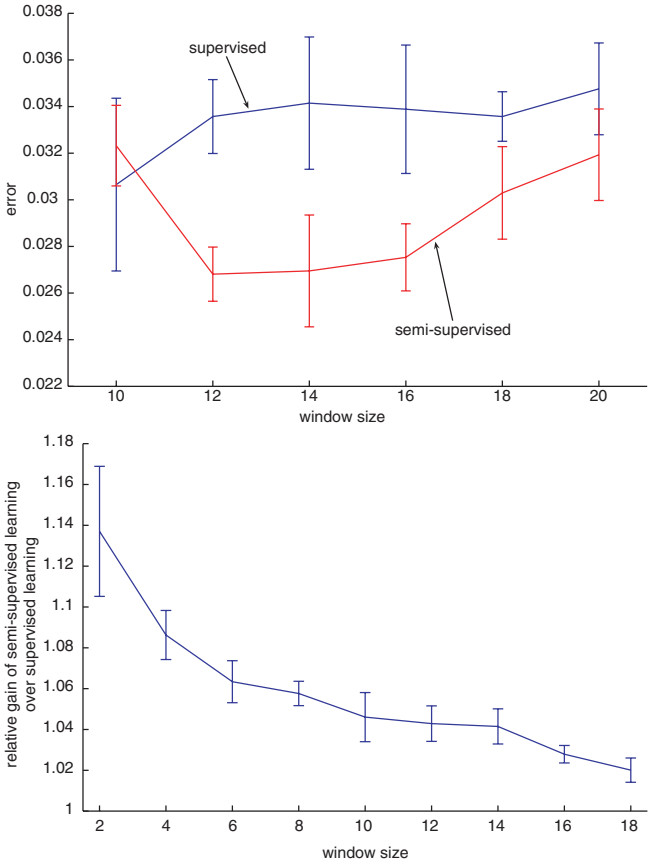


Fig. 10. Top: cross-validation error of the alignment. Bottom: The gain of semi-supervised learning over supervised learning.

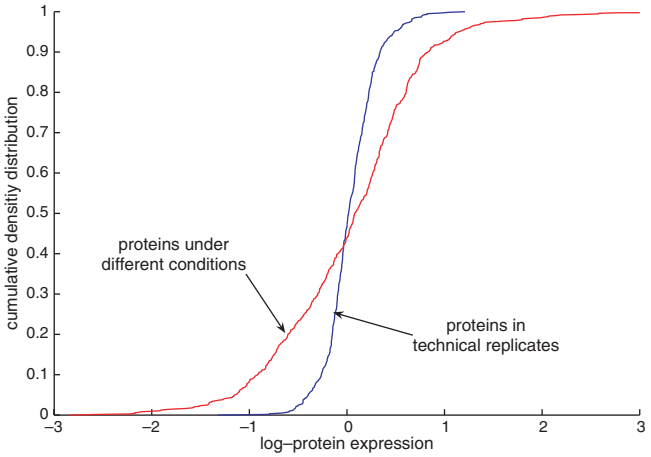


Fig. 11. Cumulative distribution function of the protein ratios for replicate measurements and for different conditioned samples.

variants. These two curves nicely summarize the benefits of the inclusion of unlabeled data: the maximum is higher (which is obviously desirable), and it is attained at smaller window sizes, which is also desirable, since it yields better localization in the mass-retention time space.

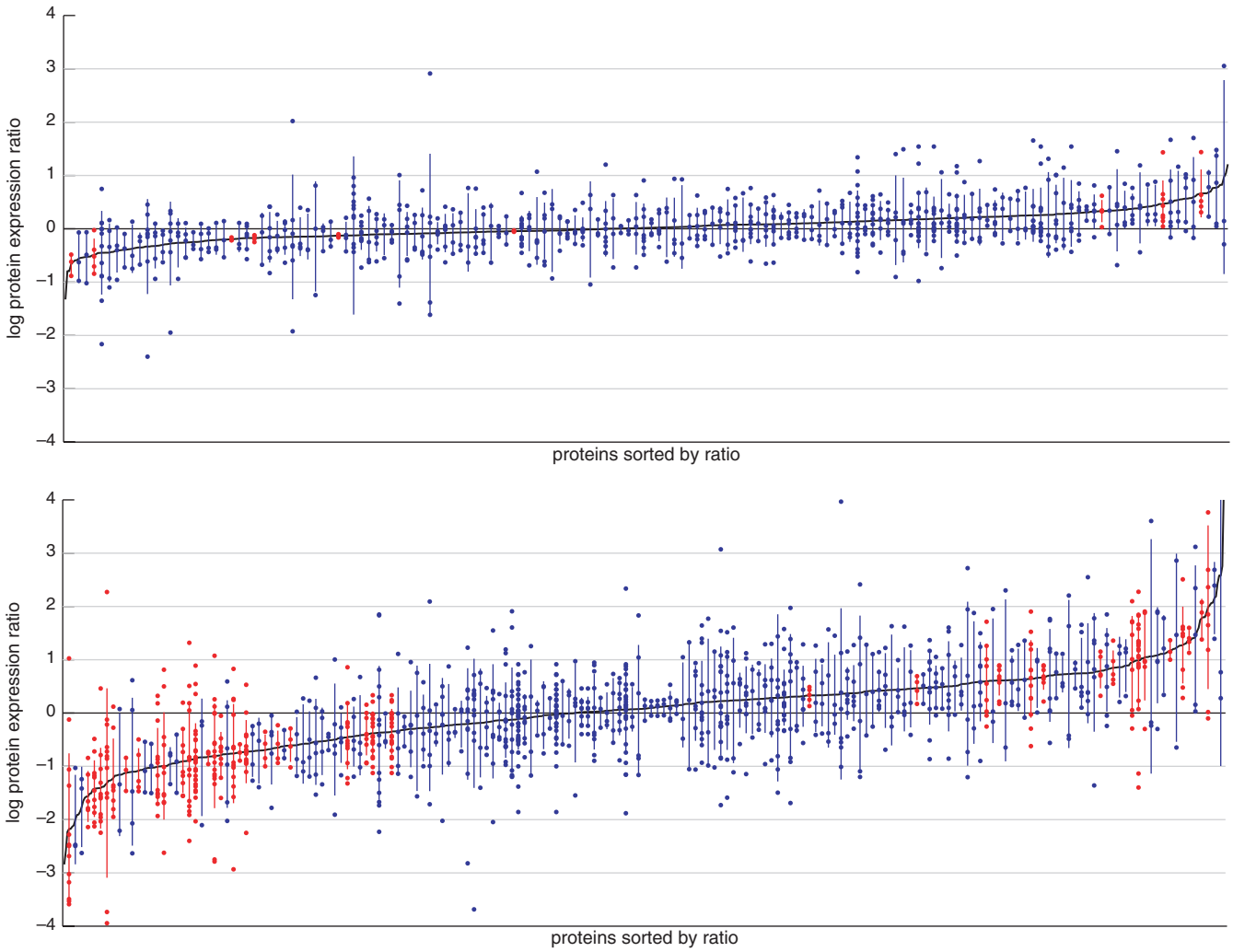


Fig. 12. Top Protein ratios for replicate measurement. Bottom: Protein ratios for different biologically conditioned samples.

3.4 Cross-validation

To show the efficiency of our approach we test it by cross-validation for **biomarker discovery**. For each alignment we divide the set of the known correspondences in a training set and a test set with size proportions (75%/25%). Only the training set is used as supervision information during alignment. On the test set we evaluate the error of the alignment. Such an error occurs, if the known sequences are assigned to different peaks. Figure 10 (top) depicts the cross-validation error for sequenced peaks. The error is plotted against the window size of acceptance for peak correspondences. An error of less than 0.03 is achieved for window sizes smaller than 15. Window sizes smaller than 10 are excluded from the plot, because the corresponding error bars are extremely large, since only very few identifications could be found. Compared to the fraction of contradicting peaks in Figure 8, the error rate on the subset of sequenced peaks is much smaller. The reason for this reduced error is that many of the contradicting peaks in the *un*supervised setting are not counting for an error in this supervised setting: a contradiction in the unsupervised setting occurs, if two different peaks from sample 1 are assigned to the same peak in sample 2. In the supervised setting such an inconsistency produces an error only if both peaks from sample 1 are *differently sequenced*. Sometimes the peak picking algorithm finds two peaks where only one peak should be placed. The sequenced MS/MS spectrum, however, is only assigned to *one* of the two ''pseudo''-peaks. In the unsupervised setting, such a situation would be treated as a contradiction, whereas in the supervised setting no error occurs.

On the bottom of the figure the gain of the semi-supervised method is plotted. We defined the gain as the ratio of confirmed correspondences for semi-supervised learning compared to supervised learning. One achieves 5% more assignments with semi-supervised learning than supervised learning at a window size of 15. Here again the improvement due to the semi-supervised method increases with smaller window size.

4 DIFFERENTIAL PROTEIN EXPRESSION

The first step towards biomarker discovery requires to compute a list of differential protein expression values. To increase the number of

sequenced peptides in each LC/MS image, we generate three replicate LC/MS/MS measurements per condition. To compare two differently conditioned samples, we compute pairwise alignments of all six LC/MS images and predict the retention time of the peptides that have been sequences from each LC/MS image to all others. This procedure yields an extensive increment in the number of sequenced peptides in each single LC/MS image. For each protein we obtain a collection of differential peptide measurements, from which the log protein ratio is estimated according to eq. (2).

To demonstrate the possibility to derive differential quantitative measurements from biological samples, we estimate the expression ratio both for replicate measurements and for differently conditioned samples. Figure 12 shows the differential protein expression. For better visualization, only a (randomly drawn) subsample of the proteins is plotted. Each protein corresponds to one column. The dots on the columns depict the differential peptide measurements. The vertical lines indicate one standard deviation. A t-test with a significance level of 0.03 rates 3.9% (24 out of 610) of the peptides as significantly over- or underexpressed for the replicate measurements. The significantly over-/under expressed proteins are colored red. For the biologically different samples (bottom panel) one can detect 24.5% (165 out of 735) of the proteins as significantly under- or overexpressed. These six times higher rate of significantly different expression levels between biologically different samples and technical replicates demonstrate that our statistical analysis is sensitive to changes in conditions. We conclude that we are able to recognize differences in protein expression by label-free differential quantitative proteomics. To conclude that the differences are caused by the different conditions, one should still compare the result with biological replicates.

5 DISCUSSION AND CONCLUSION

In the recent years the use of LC/MS measurements has received considerable attention for high-throughput analysis of proteomes. For *quantitative* differential measurements it is commonly accepted that isotopic labeling techniques such as ICAT are needed for a reliable quantitative comparison of two protein samples. These labeling techniques are not ideal, however, because they require a significantly increased complexity of the experimental setup and the necessity to mix the two labeled samples from different biological conditions. The latter is particularly problematic in applications like *biomarker discovery* where one would like to treat samples from different biological conditions separately in order to avoid a time-consuming and costly re-analysis of the, e.g., disease-specific reference sample.

As an alternative approach, we consider a *label-free* setting for comparative proteomics. The absence of isotopic labels that could guide the search for correspondences, however, imposes a severe *alignment problem* between the elements of the two samples from different biological conditions. Current approaches to solve this problem try to find alignments solely on the basis of the observed LC/MS measurements while ignoring potentially relevant additional information from the underlying sequences. In contrast to these approaches, we propose to use *tandem mass spectrometry* to extract partial sequence information of the peptides contained in the samples. Based on this subset of sequenced peptides, we compute a ''seed'' alignment by estimating a *nonlinear robust regression* function which warps one time scale into the other. Within a *semi-supervised* learning framework, this seed alignment is iteratively refined by successively including the mass peaks for which no sequence information is available. By assessing the self-consistency of the time warping in both directions, we have shown that this refinement process significantly improves the quality of the alignment.

In a large-scale experiment we have demonstrated that our method is capable of aligning *highly complex* protein samples, even if they exhibit *large variations* due to different biological conditions. It is possible to *reliably discriminate* between technical replicates and truly different biological conditions. We conclude that the proposed method bridges the gap between statistical data analysis and label-free quantitative differential proteomics.

ACKNOWLEDGEMENTS

This research was supported by the Functional Genomics Center Zurich and it was partially funded by the Swiss Initiative in Systems Biology (SystemsX: CC-SPMD and C-MOP) and by ETH-grants TH-5/04-3 and TH-41/02-2.

REFERENCES

[1] Box,G.E.P. and Tiao,G.C. *Bayesian Inference in Statistical Analysis*. Wiley, New York, 1992.

[2] Eng,J.K., McCormack,A.L. and J.R. Yates III. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Am. Soc. for Mass Spectrometry*, **5**(11), 976–989, 1994.

[3] Fischer,B., Roth,V., Roos,F., Grossmann,J., Baginsky,S., Widmayer,P., Gruissem,W., and Buhmann,J.M. NovoHMM: A hidden markov model for de novo peptide sequencing. *Anal. Chem.*, **77**(22), 7265–7273, 2005.

[4] Fox,J. *Applied Regression, Linear Models, and Related Methods*. Sage, 1997.

[5] Gygi,S.P., Rist,B., Gerber,S.A., Turecek,F., Gelb,M.H. and Aebersold,R. Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nat. Biotech.*, **17**, 994–999, 1999.

[6] Han,D.K., Eng,J., Zhou,H. and Aebersold,R. Quantitative profiling of differentiation-induced microsomal proteins using isotope-coded affinity tags and mass spectrometry. *Nat. Biotech.*, **19**, 946–951, 2001.

[7] Hirosawa,M., Hoshida,M., Ishikawa,M. and Toya,T. Masqot: multiple alignment system for protein sequences based on three-way dynamic programming. *Comp. App. in the Bioscience*, **9**(2), 161–167, 1993.

[8] Huber,P.J. *Robust Statistics*. Wiley, New York, 1981.

[9] Jacobs,J.M., Adkins,J.N., Qian,W.-J., Liu,T., Shen,Y., D.G. Camp II, and Smith,R.D. Utilizing human blood plasma for proteomic biomarker discovery. *J. of Proteome Res.*, **4**, 1073–1085, 2005.

[10] Keller,A., Nesvizhskii,A.I., Kolker,E. and Aebersold,R. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.*, **74**, 5383–5392, 2002.

[11] Listgarten,J., Neal,R.M., Roweis,S.T. and Emili,A. Multiple alignment of continuous time series. In *NIPS 17*, pages 817–824, 2005.

[12] Vest Nielsen,N.-P., Carstensen,J.M. and Smedsgaard,J. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *J. of Chromatography A*, **805**, 17–35, 1998.

[13] Nigam,K. and Ghani,R. Analyzing the effectiveness and applicability of co-training. In *CIKM '00*, pages 86–93, 2000.

[14] Tibshirani,R., Hastie,T., Narasimhan,B., Soltys,S., Shi,G., Koong,A. and Le,Q.-T. Sample classification from protein mass spectrometry, by peak probability contrasts. *Bioinformatics*, **20**, 3034–3044, 2004.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e141–e149
doi:10.1093/bioinformatics/btl223

Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE

Barrett C. Foat[1], Alexandre V. Morozov[2] and Harmen J. Bussemaker[1,3,*]

[1]Department of Biological Sciences, Columbia University, New York, NY 10027, USA, [2]Center for Studies in Physics and Biology, The Rockefeller University, New York, NY 10021, USA and [3]Center for Computational Biology and Bioinformatics, Columbia University, New York, NY 10032, USA

ABSTRACT

Motivation: Regulation of gene expression by a transcription factor requires physical interaction between the factor and the DNA, which can be described by a statistical mechanical model. Based on this model, we developed the MatrixREDUCE algorithm, which uses genome-wide occupancy data for a transcription factor (*e.g.* ChIP-chip) and associated nucleotide sequences to discover the sequence-specific binding affinity of the transcription factor. Advantages of our approach are that the information for all probes on the microarray is efficiently utilized because there is no need to delineate "bound" and "unbound" sequences, and that, unlike information content-based methods, it does not require a background sequence model.

Results: We validated the performance of MatrixREDUCE by inferring the sequence-specific binding affinities for several transcription factors in *S. cerevisiae* and comparing the results with three other independent sources of transcription factor sequence-specific affinity information: (i) experimental measurement of transcription factor binding affinities for specific oligonucleotides, (ii) reporter gene assays for promoters with systematically mutated binding sites, and (iii) relative binding affinities obtained by modeling transcription factor-DNA interactions based on co-crystal structures of transcription factors bound to DNA substrates. We show that transcription factor binding affinities inferred by MatrixREDUCE are in good agreement with all three validating methods.

Availability: MatrixREDUCE source code is freely available for non-commercial use at http://www.bussemakerlab.org/. The software runs on Linux, Unix, and Mac OS X.

Contact: Harmen.Bussemaker@columbia.edu

1 INTRODUCTION

The sequence-specific regulatory activity of a transcription factor (TF) is the result of energetically favorable interactions between the amino acids exposed in the DNA binding domain and portions of nucleic acid bases exposed in the grooves of the DNA. A computational method for discovering the binding specificity of a TF cannot provide a quantitative description of TF binding unless it considers the physical underpinnings of the TF-DNA interaction. Most physically motivated computational methods discover over-represented patterns in a set of nucleotide sequences that are considered to be bound by the TF (for review see Stormo, 2000). These methods use the information content of nucleotide patterns as a proxy for the free energy contributions of the bases found in the TF binding site (Berg and von Hippel, 1987; Stormo and Fields, 1998). Other computational methods infer physically-based TF binding specificities from measured TF binding affinities for a small set of oligonucleotides (Liu and Clarke, 2002) or from structural modeling of protein-DNA interaction (Paillard and Lavery, 2004; Endres *et al*., 2004; Morozov *et al*., 2005). However, genome-scale, quantitative measurements of TF occupancies of intergenic regions are now available due to the advent of *in vivo* chromatin immunoprecipitation microarrays (Ren *et al*., 2000; Iyer *et al*., 2001; Lieb *et al*., 2001; Simon *et al*., 2001; Lee *et al*., 2002; Harbison *et al*., 2004), *in vitro* protein binding microarrays (PBM; Mukherjee *et al*., 2004), and DNA immunoprecipitation microarrays (DIP-chip; Liu *et al*., 2005). Thus, it is no longer necessary to rely on small data sets, availability of protein-DNA structures, or the analogy between information content and statistical mechanics to infer free energy representations of transcription factor binding sites.

We have developed a method, implemented as the program MatrixREDUCE (Foat *et al*., 2005), that infers the sequence specificity of a TF directly and accurately from genome-wide TF occupancy data by fitting a statistical mechanical model for TF-DNA interaction (Figure 1). The sequence specificity of the TF's DNA-binding domain is modeled using a position-specific affinity matrix (PSAM), representing the change in the binding affinity (K_d) whenever a specific position within a reference binding sequence is mutated. To validate the physical model of MatrixREDUCE, we discovered the PSAMs for several TFs in *S. cerevisiae* and compared the results with three other independent sources of TF sequence-specific affinity information: (i) experimentally measured K_d's as determined by *in vitro* methods (Gailus-Durner *et al*., 1996; Liu and Clarke, 2002; Pierce *et al*., 2003), (ii) *lacZ* reporter assays for promoters with systematically mutated binding sites (Gailus-Durner *et al*., 1996; Pierce *et al*., 2003), and (iii) relative K_d's obtained by using a physical model of protein-DNA interaction that makes binding affinity predictions starting from a co-crystal structure of the protein-DNA complex (Morozov *et al*., 2005). We find a surprising level of agreement between MatrixREDUCE-predicted TF binding affinities, experimental measurements, and structural predictions, suggesting that MatrixREDUCE is a

*To whom correspondence should be addressed.

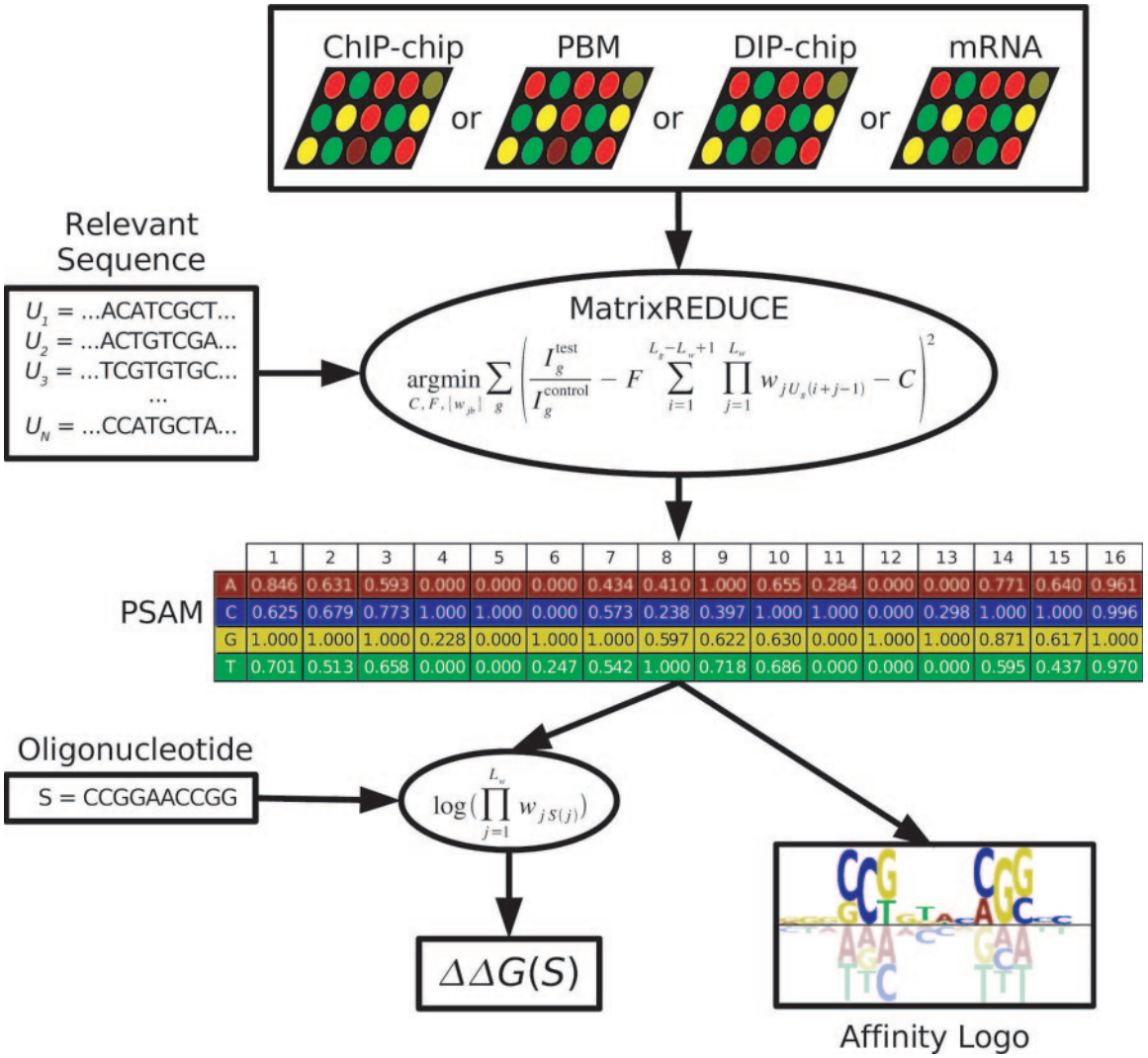


Fig. 1. The flow of data. A microarray measurement of TF occupancies (ChIP-chip, PBM, DIP-chip, or differential mRNA expression data) and relevant nucleotide sequences for each microarray feature are used as input to MatrixREDUCE. MatrixREDUCE performs a least-squares fit to a statistical-mechanical model of TF-DNA interaction to discover the relative contributions to the free energy of binding for each nucleotide at each position in the generalized TF binding site. These contributions are represented as a position specific affinity matrix (PSAM) containing the relative equilibrium constants of the TF-DNA interaction, with the highest affinity nucleotide at each position scaled to a value of one ($\Delta\Delta G = 0$). The PSAM can be converted into an affinity logo that graphically represents the $\Delta\Delta G$'s for each nucleotide at each position relative to the average $\Delta\Delta G$ at the respective positions. The PSAM can also be used to predict the relative TF occupancy of any nucleotide sequence, allowing the PSAMs inferred by MatrixREDUCE to be compared with experimental measurements of TF binding affinities for particular oligonucleotides.

powerful and accurate tool for the elucidation of physically accurate TF sequence-specific binding affinities.

2 RELATED WORK

In contrast to information theory-based methods of defining nucleotide-binding protein specificities, MatrixREDUCE belongs to a small but growing class of methods that infer binding affinities by directly fitting a physical model to experimental data. The first such method was introduced by Stormo *et al*. (1986) who noted, ''When quantitative data are known for many sequences one can solve for the matrix elements that give the best fit between the sequences and those data.'' For Stormo *et al*. (1986) the quantitative data were β-galactosidase activities for genes containing mutated binding sites of the *E. coli su*2 amber stop codon suppressor. A similar type of analysis was performed by Liu and Clarke (2002) who fit a physical model for transcription factor binding to electrophoretic mobility shift assay (EMSA) data measuring affinity of the *S. cerevisiae* Leu3 TF for several oligonucleotides. The physical model behind MatrixREDUCE is the same as that employed by Stormo *et al*. (1986) and Liu and Clarke (2002). However, our ''quantitative data'' are microarray probe intensities, which measure TF occupancy over long chromosomal regions with unknown binding site locations. Thus, the MatrixREDUCE model integrates the binding signal over the entire length of the sequence. The GOMER method of Granek and Clarke (2005) performs a similar

integration of signal over long regulatory sequences relevant to measured microarray intensities. However, GOMER was only used to test hypotheses about the regulatory mechanisms of TFs for which a binding site weight matrix had already been defined by other methods. Granek and Clarke (2005) did not attempt to fit the GOMER model directly to experimental data to infer the binding affinities of TFs. Finally, also of note are the QPMEME algorithm of Djordjevic *et al*. (2003) and the work of Djordjevic and Sengupta (2006) which use maximum likelihood procedures to infer PSAMs by fitting physical models to known TF binding sites and SELEX data, respectively, but rely on the prior delineation of ''bound'' sequences.

3 METHODS

3.1 Modeling TF-DNA interaction

We will develop the statistical-mechanical model used by MatrixREDUCE starting with a transcription factor P that binds to a DNA sequence S to form the TF-DNA complex PS:

$$P + S \underset{k_{\text{off}}}{\overset{k_{\text{on}}}{\rightleftarrows}} PS \tag{1}$$

The affinity of the TF for the sequence can be expressed in terms of its equilibrium dissociation constant $K_d(S)$:

$$K_d(S) = \frac{[P][S]}{[PS]} = \frac{k_{\text{off}}}{k_{\text{on}}} = e^{\Delta G/RT}, \tag{2}$$

which is directly related to ΔG, the Gibbs free energy of binding per mole (R is the gas constant and T is temperature). The occupancy $N(S)$ of sequence S by transcription factor P can be expressed as the concentration of TF-DNA complex divided by the total concentration of DNA (bound or unbound):

$$N(S) = \frac{[PS]}{[PS] + [S]} = \frac{[P]}{[P] + K_d(S)}. \tag{3}$$

For simplicity, we will assume that the TF concentration $[P]$ is much smaller than $K_d(S)$. This assumption seems physiologically plausible because in this regime, the highest affinity binding sites in the genome will be the most responsive to a change in the nuclear concentration of active TF. Thus, the occupancy becomes:

$$N(S) \approx \frac{[P]}{K_d(S)} = [P]K_a(S), \tag{4}$$

where

$$K_a(S) \equiv K_d^{-1}(S). \tag{5}$$

Consider a single point mutation from the original reference sequence S_{ref} to base b at position j resulting in the mutated sequence S_{mut}. Such a mutation will give rise to an additive change $\Delta\Delta G$ in the free energy of binding or, equivalently, a multiplicative change w_{jb} in $K_a(S_{\text{ref}})$:

$$w_{jb} = \frac{K_a(S_{\text{mut}})}{K_a(S_{\text{ref}})} = e^{\Delta\Delta G/RT}, \tag{6}$$

where

$$\Delta\Delta G = \Delta G(S_{\text{ref}}) - \Delta G(S_{\text{mut}}). \tag{7}$$

To be able to generalize the binding of transcription factor P to a sequence S_{mut} with more than one point mutation, we assume that the free energy contributions for each position in the binding site are independent (Benos *et al*., 2002) and therefore additive. Equivalently, we can multiply the w_{jb}'s for any nucleotide sequence to obtain the overall $K_a(S_{\text{mut}})/K_a(S_{\text{ref}})$ ratio. Thus, the occupancy of a particular binding site S_{mut} of length L_w with nucleotide sequence $S_{\text{mut}}(1,2,\ldots,L_w) = (b_1, b_2, \ldots, b_{L_w})$ is:

$$N(S_{\text{mut}}) = [P]K_a(S_{\text{ref}}) \prod_{j=1}^{L_w} w_{jS_{\text{mut}}(j)}. \tag{8}$$

The occupancy $N(U_g)$ for the entire promoter region U_g of gene g equals the sum of occupancies for each binding site window of length L_w at each position i over the length L_g of the sequence U_g:

$$N(U_g) = [P]K_a(S_{\text{ref}}) \sum_{i=1}^{L_g - L_w + 1} \prod_{j=1}^{L_w} w_{jU_g(i+j-1)}, \tag{9}$$

where $U_g(i)$ is the base at position i in sequence U_g.

3.2 Modeling genome-wide TF occupancy data

Recent technologies such as ChIP-chip (Ren *et al*., 2000; Iyer *et al*., 2001; Lieb *et al*., 2001; Simon *et al*., 2001; Lee *et al*., 2002; Harbison *et al*., 2004), PBM (Mukherjee *et al*., 2004), and DIP-chip (Liu *et al*., 2005) provide indirect but quantitative information about the TF occupancy of large genomic regions. For each segment of DNA there are two microarray intensities. The test intensity I_g^{test} is equal to a background intensity α^{test} plus a term that, to first approximation, is proportional (γ) to the occupancy $N(U_g)$ by the TF, either because the amount of TF bound to the probe contributes directly to the signal intensity (PBM) or because it determines the proportion at which an immunoprecipitated TF-DNA fragment is present in the sample (ChIP-chip or DIP-chip). The control intensity I_g^{control} is only the result of background signal α^{control}. Allowing for experimental noise ϵ_g, we obtain:

$$\frac{I_g^{\text{test}}}{I_g^{\text{control}}} = \frac{\gamma N(U_g) + \alpha^{\text{test}}}{\alpha^{\text{control}}} + \epsilon_g \equiv \beta N(U_g) + C + \epsilon_g \tag{10}$$

Using Equation 9 for the occupancy $N(U_g)$, we obtain:

$$\frac{I_g^{\text{test}}}{I_g^{\text{control}}} = F \sum_{i=1}^{L_g - L_w + 1} \prod_{j=1}^{L_w} w_{jU_g(i+j-1)} + C + \epsilon_g, \tag{11}$$

where

$$F = \beta[P]K_a(S_{\text{ref}}). \tag{12}$$

Note that β, $[P]$, and $K_a(S_{\text{ref}})$ cannot be determined separately without additional information such as the real protein concentration or $K_a(S_{\text{ref}})$.

MatrixREDUCE discovers the set of w_{jb} elements as well as F and C by performing a least squares fit to the measured intensity ratios:

$$(C, F, \{w_{jb}\}) = \underset{C, F, \{w_{jb}\}}{\operatorname{argmin}} \sum_g \left(\frac{I_g^{\text{test}}}{I_g^{\text{control}}} - F \sum_{i=1}^{L_g - L_w + 1} \prod_{j=1}^{L_w} w_{jU_g(i+j-1)} - C \right)^2. \tag{13}$$

The $4 \times L_w$ matrix of K_a ratios w_{jb} ($3L_w$ parameters plus L_w reference nucleotide values) for all nucleotides at all positions in the binding site is referred to as the position specific affinity matrix (PSAM). Each position j in the PSAM is rescaled such that the largest w_{jb} is equal to unity, without loss of generality.

Differential mRNA expression microarray data, which measures the change in mRNA concentrations in cells from two different experimental conditions, can be used in place of genome-wide TF occupancy data. This substitution is reasonable since, to first approximation, the transcription rate of genes is proportional to the total TF occupancy along the associated promoter regions. Genome-wide occupancy data is preferable, however, since it is a more direct measure of TF-DNA interaction and since the design of the experiments provides the TF identities for the discovered PSAMs.

3.3 MatrixREDUCE implementation and parameters

MatrixREDUCE was implemented in Perl and C as outlined above and as previously described (Foat *et al*., 2005) with some modifications. Briefly, MatrixREDUCE takes microarray intensities and corresponding

nucleotide sequence data as input. It first finds a gapped dyad motif (*e.g.* Leu3: CCG-4nt-CGG), out of all possible dyad motifs of a fixed number of nucleotides and a range of gap sizes, whose occurrences best correlate with the measured intensities for the same sequences. The best dyad motif is then converted into a seed matrix by filling in the gap with N's and extending out a user defined number of flanking N's on either side of the best-scoring dyad. In the $4 \times L_w$ seed matrix, acceptable nucleotides (all nucleotides for N's, a single specific nucleotide at positions within the top scoring motif) are given K_a ratios of one and unacceptable nucleotides are given a very small K_a ratio $w_{\min}$. This seed matrix serves as the starting point for a quasi-Newton numerical minimization of Equation 13 to find the optimal PSAM. The new version of MatrixREDUCE uses a *k*-fold cross-validation to determine the significance of each discovered PSAM. After converging on a PSAM, the input data is split into *k* random subsets of array features with associated sequences. The optimal PSAM is then used to seed each of *k* re--optimizations of the PSAM. A *t*-value (Pearson correlation) for the goodness of fit is calculated for the optimal PSAM of each subset. Finally, the *P*-value corresponding to the average *t*-value for the *k* re-optimizations is used to test whether the originally optimized PSAM should be kept. This procedure does not test the significance of the optimal PSAM itself, but rather it tests whether the data contains widely distributed, explainable variance. Thus, false PSAMs due to a few outliers are prevented. While not relevant to the current study, MatrixREDUCE can iteratively build a linear model of multiple PSAMs that best explain a particular data set (see Foat *et al*., 2005).

The parameters for the runs of MatrixREDUCE were as follows: For all runs, the length of each of the two dyads of the seed motifs was three, the length of the added flanks on each side of the dyad was three, the minimum gap was zero, the *k* cross-validations were two, and $w_{\min}$ was 10^{-5}. For all runs on ChIP-chip and PBM data, the maximum acceptable *P*-value was 10^{-3} and the maximum dyad gap was twenty. For all runs on DIP-chip data, the maximum acceptable *P*-value was 10^{-6} and the maximum dyad gap was ten. For all runs on differential mRNA expression data, the maximum acceptable *P*-value was 10^{-3} and the maximum dyad gap was eleven.

3.4 Microarray and sequence data

All microarray data was gathered from publication supplements. We chose specific TFs to analyze based on the availability of experimental K_d data or crystal structure data. PSAMs were inferred by MatrixREDUCE for chromatin immunoprecipitation microarrays (ChIP-chip) using the microarray data and microarray feature sequences from Harbison *et al*. (2004). These ChIP-chip experiments were performed under a variety of culture conditions, including rich media (YPD); sulfometuron methyl (SM), an inhibitor of amino acid biosynthesis; and treatment with rapamycin (RAPA). PSAMs were inferred for PBM experiments using the microarray data from Mukherjee *et al*. (2004) and the feature sequence data from Harbison *et al*. (2004) as the two studies used the same array features. PSAMs were inferred for Leu3 using the DIP-chip microarray data and feature sequences from Liu *et al*. (2005). Liu *et al*. (2005) performed DIP-chip experiments using two different concentrations of Leu3, 4nM and 40nM, and PSAMs were inferred for each concentration. The PSAM for Ndt80 was inferred from differential mRNA expression microarray data measuring the sporulation response in a *ndt*80 deletion strain versus a wild-type strain (Chu *et al*., 1998). The sequence data for the Ndt80 PSAM inference was the 800 bp upstream of every yeast gene, retrieved from the *Saccharomyces* Genome Database (Issel-Tarver *et al*., 2002) and purged of redundant sequences as previously described (Foat *et al*., 2005). All microarray intensities were analyzed as the ratio of the experimental sample intensity to the control sample intensity with the exception of the *ndt*80 deletion data, which was analyzed as the $\log_2$-ratio. All microarray data was purged of extreme outliers before being analyzed by MatrixREDUCE (Grubbs' test, *P*-value $= 10^{-10}$; Grubbs, 1969).

3.5 Gel shift and *lacZ* expression data

While prone to their own inaccuracies, experimentally measured *in vitro* binding affinities and changes in *lacZ* expression served as our ''gold standards'' to assess the validity of our MatrixREDUCE model. The electrophoretic mobility shift assay (EMSA) is able to provide direct estimates of K_d's for a TF binding to particular oligonucleotides (Fried and Crothers, 1981). The ratio of the EMSA-measured K_d of a reference oligonucleotide S_{ref} to the K_d of one of the other tested oligonucleotides S_{mut} provides the same information as the product across the MatrixREDUCE PSAM over the same sequence for the same TF. In the simplifying scenario where the length of the oligonucleotides is the same as the length L_w of the PSAM, we have

$$\frac{K_d(S_{\text{ref}})}{K_d(S_{\text{mut}})} = \prod_{j=1}^{L_w} w_{jS_{\text{mut}}(j)}. \tag{14}$$

While the biological processes involved are considerably more complex, *lacZ* expression data can be employed to the same end. If we assume that β-galactosidase activity, concentration of β-galactosidase, the amount of mRNA expressed, the specific recruitment of RNA polymerase to the promoter, and the promoter occupancy by the TF are all proportional to each other, then relative K_d's are reflected in the ratio of β-galactosidase activities between the assay using the reference binding site and another assay using a different binding site. Thus, we used *lacZ* reporter expression assay data in a similar manner to EMSA-derived K_d data to confirm the results of MatrixREDUCE.

Experimentally determined *in vitro* binding affinities and *lacZ* reporter expression activity data were gathered from publications. The K_d data and *lacZ* expression data for Abf1 are from Gailus-Durner *et al*. (1996); K_d data for Leu3 are from Liu and Clarke (2002); and K_d data and *lacZ* expression data for Ndt80 and Sum1 are from Pierce *et al*. (2003).

To compare the experimental K_d measurements with MatrixREDUCE PSAMs, all experimental K_d and *lacZ* expression data was first converted to K_a ratios by normalizing with respect to the value of the highest affinity oligonucleotide. The K_a ratios were then log-transformed to obtain the $\Delta\Delta G$ values. MatrixREDUCE PSAMs for each TF were converted to $\Delta\Delta G$'s relative to the highest affinity oligonucleotide from the respective experiment. The sum of the $\Delta\Delta G$ values was calculated for the best PSAM-matching window in each of the experimentally tested sequences. If a sequence was shorter than the PSAM, the sum was taken over only the best matching positions within the PSAM. All experimental $\Delta\Delta G$'s were then compared to the PSAM $\Delta\Delta G$'s by plotting and by calculating Pearson correlations.

BioProspector (Liu *et al*., 2001) and MDscan (Liu *et al*., 2002) are popular information theory-based methods for determination of TF binding specificities. To compare the quality of the results from these methods with MatrixREDUCE results, position-specific scoring matrices (PSSMs) were derived from BioProspector and MDscan outputs by calculating the frequencies of each base at each position in the putative binding sites and then dividing by a background frequency for each respective base. Two different background frequencies were tested: equal nucleotide probabilities and nucleotide probabilities for intergenic sequences in *S. cerevisiae*. Once the PSSMs had been created, they were tested against experimental EMSA and *lacZ* data in the same manner as the MatrixREDUCE PSAMs above.

3.6 Structural modeling

DNA binding affinities and specificities of TFs are determined by the forces of electrostatics, solvation, the hydrogen bonding patterns, and shape complementarity at the binding interface. The magnitude of these contributions to the binding free energy can in principle be calculated given a structure of the protein bound to its cognate DNA site. Therefore, it should be possible to predict PSAMs starting from the experimentally available structure of the protein-DNA complex (solved by either X-ray diffraction or NMR), or, in the absence of the exact structure, from a suitably constructed homology model. Under the assumption that the base pair energies contribute approximately independently to the total binding affinity (Benos *et al*., 2002), all one-point base pair mutations are introduced into the DNA binding site. Protein-DNA binding energies $\Delta G =$

$G_{\text{prot-dna}} - G_{\text{prot}} - G_{\text{dna}}$ are then evaluated for each mutation. Mutations in the reference binding site result in changes of protein-DNA binding energies ($\Delta\Delta G$; Equation 7). A table of $\Delta\Delta G$ values can be used to construct a PSAM that is directly comparable with MatrixREDUCE predictions.

We have previously developed two alternative approaches for predicting TF binding affinities and specificities starting from the protein-DNA structure (Morozov *et al*., 2005). In one approach, the ''all atom model'' (which builds on the ROSETTA protein-nucleic acid interaction model of Havranek *et al*., 2004), both direct and indirect readout mechanisms contribute to the recognition of the DNA binding site: $\Delta G = \Delta G_{\text{direct}} + \Delta G_{\text{indirect}}$. Direct readout is mediated by protein amino acid-DNA base interactions, while indirect readout is encoded in the shape of the DNA site imparted by the bound protein, primarily through non-specific amino acid-DNA phosphate backbone contacts. Direct protein-DNA interactions are modeled as a linear combination of the repulsive and attractive parts of the Lennard-Jones potential, the orientation-dependent hydrogen bonding potential (Kortemme *et al*., 2003), and the Generalized Born electrostatics and solvation model (Onufriev *et al*., 2004):

$$\begin{aligned}\Delta G_{\text{direct}} = {} & w_{\text{LJrep}} E_{\text{LJrep}} + w_{\text{LJattr}} E_{\text{LJattr}} \\ & + w_{\text{hb}} E_{\text{hb}} + w_{\text{el}} G_{\text{el}},\end{aligned} \tag{15}$$

where each term is a sum over all protein-DNA and protein-protein atomic pairs, and $\{w\}$ is a set of fitting weights. Indirect readout is modeled using an effective harmonic representation of the DNA conformational energy (Olson *et al*., 1998):

$$\Delta G_{\text{indirect}} = w_{\text{dna-bp}} \sum_{\text{bp}} E^{\alpha\beta}_{\text{dna-bp}} + w_{\text{dna-bs}} \sum_{\text{bs}} E^{\alpha\beta}_{\text{dna-bs}}, \tag{16}$$

where the first sum is over all base pairs in the DNA site (α, β denote bases in a base pair), and the second sum is over all consecutively stacked base pair steps (α, β denote base pairs in a base step). Base pairs and base steps are counted once in the 5′ to 3′ direction. The first term penalizes deviations from canonical base pairing, while the second term captures base stacking energies. The quadratic energy terms are given by:

$$E^{\alpha\beta}_{\text{dna-bs/dna-bp}} = \frac{1}{2} \sum_{i=1}^{6} \sum_{j=1}^{6} f^{\alpha\beta}_{ij} \delta\theta^{\alpha}_{i} \delta\theta^{\beta}_{j}, \tag{17}$$

where the sums run over six geometric degrees of freedom θ_i (Twist, Tilt, Roll, Shift, Slide and Rise for base pair steps; Opening, Buckle, Propeller, Shear, Stretch and Stagger for base pairs; Lu and Olson, 2003). The DNA potential is a quadratic expansion in $\delta\theta_i$ (deviations of the degrees of freedom θ_i from their average values computed using a set of non-homologous protein-DNA complexes). The force constants f_{ij} are evaluated by inverting the covariance matrix of $\delta\theta_i$ obtained with the same protein-DNA dataset: $f^{-1}_{ij} = \langle \delta\theta_i \delta\theta_j \rangle$. All six weights are simultaneously fit to experimental $\Delta\Delta G$ data using a generalized linear model (implemented in the statistical software package R): $(w_{\text{LJrep}}, w_{\text{LJattr}}, w_{\text{hb}}, w_{\text{el}}, w_{\text{dna-bp}}, w_{\text{dna-bs}}) = (0.00, 0.46, 0.77, 0.27,\ 0.03, 0.03)$. No conformational flexibility is allowed at the protein-DNA interface. Further details on the fitting procedure and comprehensive tests of the all-atom free energy function can be found in Morozov *et al*. (2005).

In another approach, we developed a ''contact model'' that exploits the structure of the protein-DNA complex bound to a high affinity reference DNA sequence but does not require detailed predictions of protein-DNA interaction energies. In the contact model each mutated base in the PSAM column i incurs equal energy cost relative to the consensus base from the reference sequence:

$$\Delta\Delta G^{i}(N) = \begin{cases} E_{\max} \left[f_1(N_{\max}) \log\left(1 - N/N_{\max}\right) \right. \\ \quad \left. - f_2(N_{\max}) \log\left(1 + 3N/N_{\max}\right) \right] \ (N < N_{\max}) \\ E_{\max} \ (N \geq N_{\max}) \end{cases} \tag{18}$$

Here, N is the number of protein amino acid-DNA base atomic contacts summed over the base pair i (atomic contact is defined by a distance of less than 4.5 Å; hydrogen atoms are excluded from the counts), and $N_{\max}$ is the number of contacts above which the maximum energy penalty $E_{\max}$ is imposed. $f_1(N_{\max})$ and $f_2(N_{\max})$ are fixed prefactors defined in Morozov *et al*. (2005). $E_{\max}$ together with $N_{\max}$ constitute the free parameters of the contact model and are adjusted simultaneously to maximize the fraction of correct predictions and minimize the average error over the $\Delta\Delta G$ data set identical to that used in fitting the all-atom model. The fraction of correct predictions is based on a binary function: a prediction is considered to be correct if both computational and experimental $\Delta\Delta G$'s are less than 1.0 kcal/mol, or greater than 1.0 kcal/mol, or else separated by less than 0.3 kcal/mol. The global minimum for the fit is found by exhaustive search; the best fit is obtained with $N_{\max} = 15$, $E_{\max} = 3.0$ kcal/mol.

3.7 Affinity logos

Information content-based weight matrices are usually displayed as sequence logos (Schneider and Stephens, 1990) However, MatrixREDUCE weight matrices are discovered without a background sequence model. Thus, an appropriate logo should display the actual relative free energies of binding for each nucleotide at each position rather than information content. Therefore, we created affinity logos, which are constructed as follows: For each position in the PSAM, the average $\Delta\Delta G$ is calculated. Then, the difference between each individual $\Delta\Delta G$ and the average $\Delta\Delta G$ at that position is computed; the absolute value of this difference is the height of the character representing that nucleotide. If the difference is positive (more favorable than average), the letter is placed above a horizontal black line through the center of the logo. If the difference is negative (less favorable than average) the letter is placed below the black line. Larger letters are stacked on smaller letters moving outward from the black line. The height of the letter can be interpreted as free energy difference from the average in units of *RT*. Thus, an intuitive high amplitude is given to the nucleotide positions that most contribute to the sequence specificity of the TF. To highlight that the characters representing the high affinity nucleotides are above the black line, the characters representing the low affinity nucleotides are made partially transparent. However, maintaining the representation of the poor affinity nucleotides below the center line allows the viewer to immediately see which nucleotide substitutions are most unfavorable to binding.

3.8 PSAM to PSAM alignments and correlations

By inspection of affinity logos, one can make qualitative observations about the similarity between any two PSAMs. However, a quantitative measure of similarity allows for more objective comparisons. Before two PSAMs can be compared, they must first be aligned. Pearson correlations were calculated between the $\Delta\Delta G$ values for each nucleotide at each position for every possible overlap of the two PSAMs for both the forward and the reverse complement alignments. After the best overlap position and strand was determined from the best correlation P-value, the $\Delta\Delta G$'s of the two PSAMs were recentered relative to a common reference consensus sequence. Finally, the P-value for the Pearson correlation between the two optimally aligned and transformed PSAMs was calculated and subjected to a Bonferroni correction for the number of alignments that were tested.

4 RESULTS

4.1 PSAMs inferred by MatrixREDUCE agree well with experimental measurements of TF binding affinity

We discovered the position specific affinity matrices (PSAMs) for the *Saccharomyces cerevisiae* TFs Rap1, Ndt80, Gcn4, Leu3, Abf1, and Sum1 by applying MatrixREDUCE to genome-wide TF occupancy data and, in the case of Ndt80, differential mRNA expression microarray data (Figure 2A). Experimental measurements of relative K_d's (EMSA or *lacZ* expression) for specific oligonucleotides were available for Abf1, Leu3, Ndt80, and Sum1. EMSA has long been employed to determine the

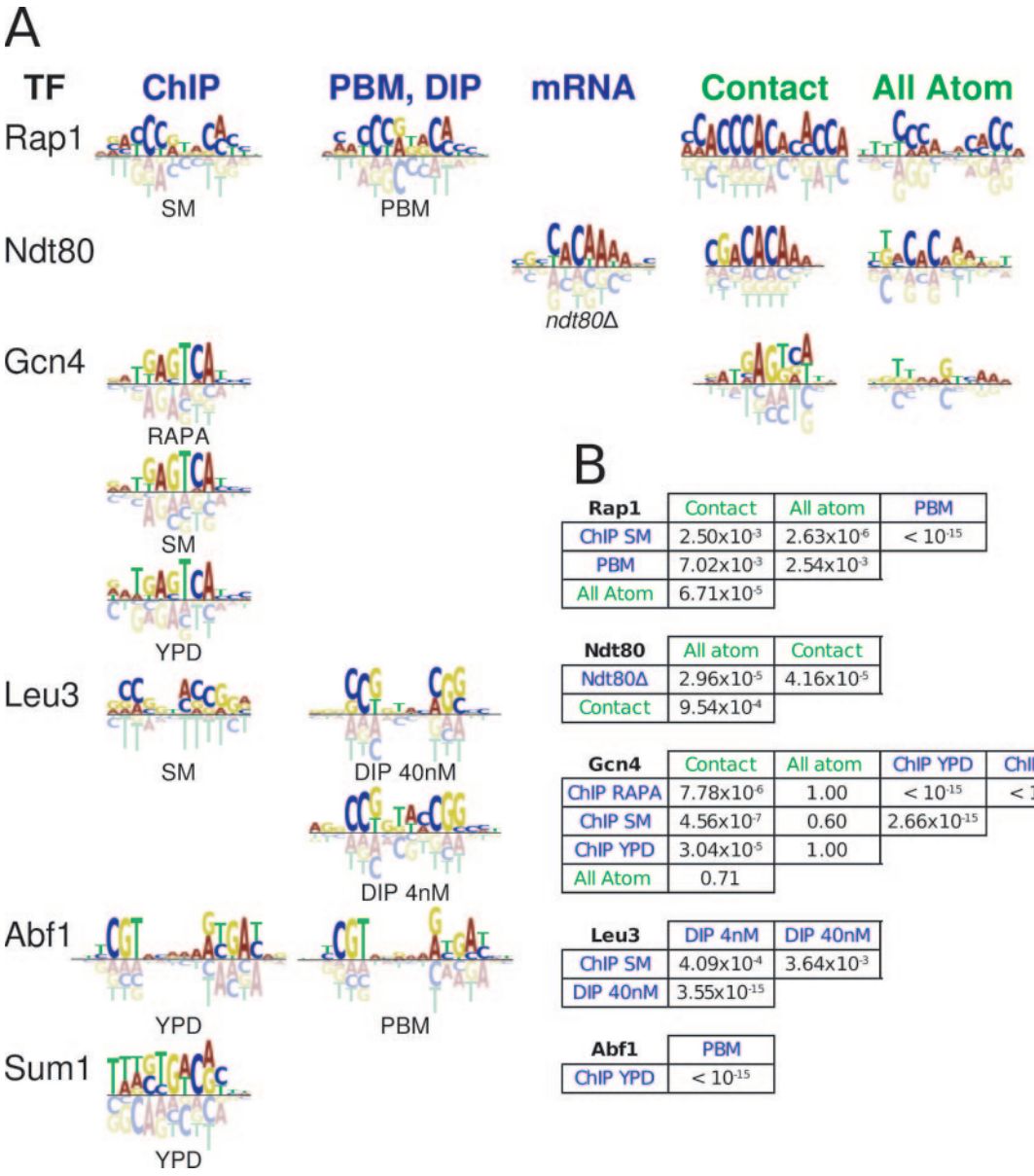


Fig. 2. Comparison of PSAMs—affinity logos and correlations. (**A**) The PSAMs represented in the columns with blue headers were inferred by Matrix-REDUCE from ChIP-chip, PBM, DIP-chip, or mRNA differential expression microarray data. YPD (rich media), SM (sulfometuron methyl), and RAPA (rapamycin) refer to the environmental conditions to which the test sample was exposed before the ChIP-chip experiment. The DIP-chip experiments were performed with two different concentrations of Leu3, 4nM and 40nM. An *ndt80* deletion (*ndt80*Δ) versus wild-type mRNA expression experiment (mRNA) was used to obtain the Ndt80 PSAM. The PSAMs represented in the columns green headers were inferred by modeling TF-DNA interactions based on crystal structures of the TFs using two different methods, a contact-only model and an all atom model. (**B**) All PSAMs for each TF were aligned pairwise and the Pearson correlation between the $\Delta\Delta G$ values of both PSAMs for the best alignment was calculated. The *P*-value for this correlation is a measure of similarity between the PSAMs. Again, blue labels indicate PSAMs inferred by MatrixREDUCE PSAMs and green labels indicate structurally inferred PSAMs.

DNA-binding affinities of TFs in vitro. Likewise, the *lacZ* reporter assay has long been used to measure the difference in activities of TF binding sites. We claim that a PSAM inferred by MatrixREDUCE from genome-wide TF occupancy data can be used to predict the relative binding affinities of the measured TF to any sequence. Therefore, EMSA and *lacZ* expression data provide nearly ideal data sets for validation of the MatrixREDUCE approach. For each combination of experimentally tested sequence, experimental method (EMSA or *lacZ*), and TF, we compared the experimental $\Delta\Delta G$ with the $\Delta\Delta G$ predicted from a PSAM for the same TF (Figure 3). In every case, the experimental $\Delta\Delta G$ values strongly correlated with the PSAM-predicted $\Delta\Delta G$ values, with R^2's ranging from 0.36 to 0.88. Thus, PSAMs inferred by MatrixREDUCE seem to be good models of the true relative DNA binding affinities of the corresponding TFs. Unexpectedly, all of the regressions of experimental $\Delta\Delta G$'s on MatrixREDUCE $\Delta\Delta G$'s have

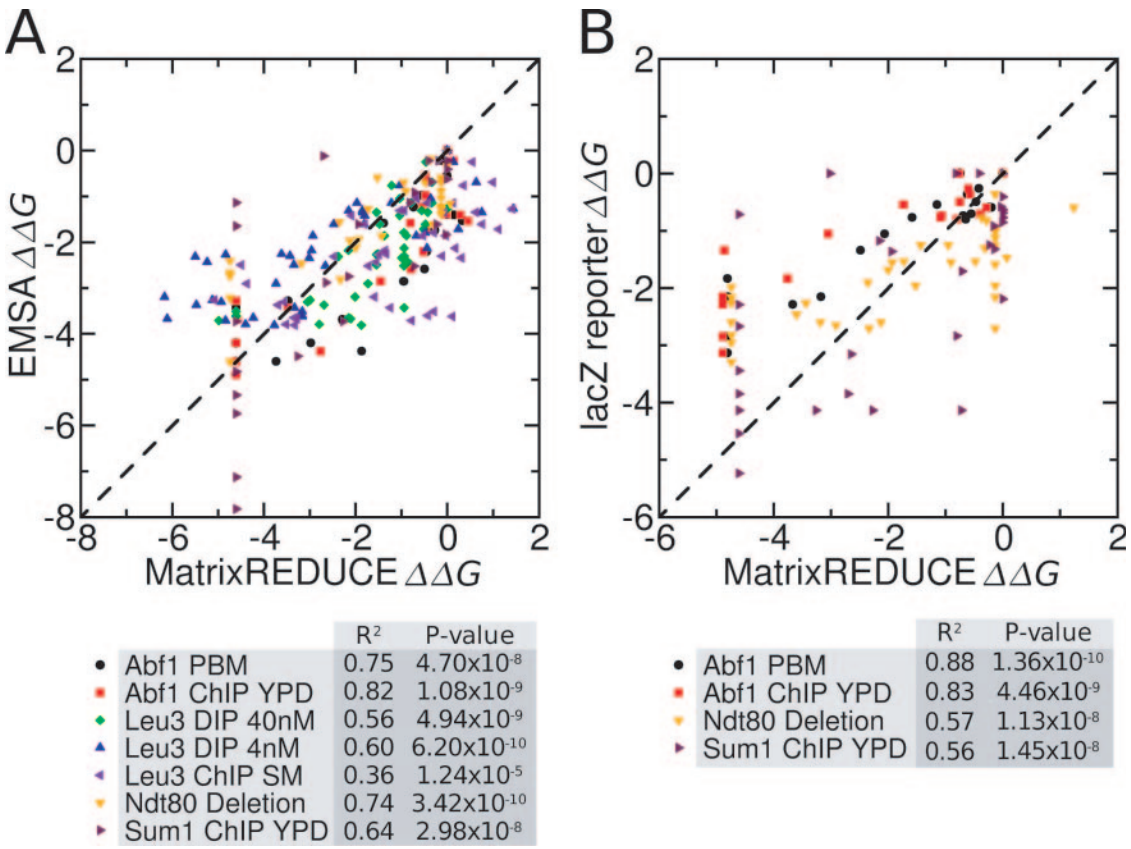


Fig. 3. Comparison of experimentally measured ΔΔG's with MatrixREDUCE PSAM-predicted ΔΔ*G*'s. Experimental measurements of ΔΔ*G*'s were derived from EMSA (**A**) and *lacZ* reporter assays (**B**). The experimental ΔΔ*G* values are plotted along the vertical axes. Predicted ΔΔ*G*'s were calculated from the PSAM for each tested TF for the same oligonucleotide sequences that were measured in each experiment. The MatrixREDUCE-predicted ΔΔ*G* values are plotted along the horizontal axes. In this representation, the higher affinity oligonucleotides have more positive ΔΔ*G*'s. The diagonal dashed line represents experimental ΔΔ*G* equal to MatrixREDUCE ΔΔ*G*. ΔΔ*G*'s are in units of *RT*, where *R* is the gas constant and *T* is the temperature. The R^2 and *P*-values for the Pearson correlations between the experimental and predicted ΔΔ*G*'s are presented for each PSAM-experimental data pair.

slopes less than one (range: 0.31 to 0.94). It seems that MatrixREDUCE produces a slightly larger range of predicted ΔΔ*G*'s than is realized in experiments. Nonetheless, the MatrixREDUCE PSAM-predicted ΔΔ*G*'s are close to the experimentally inferred ΔΔ*G*'s in most cases, especially among the highest affinity sequences.

4.2 PSAMs inferred by MatrixREDUCE agree well with PSAMs inferred by structural models

Both genome-wide TF occupancy data and crystal structures of protein-DNA complexes are available for Ndt80, Gcn4, and Rap1. Thus, we were able to compare MatrixREDUCE PSAMs with those based on *ab initio* structural models (see Methods; Figure 2A). The structurally inferred PSAMs for Ndt80 were obtained from its co-crystal structure bound to a high affinity GACACAAAA site, solved at 1.4 Å resolution (Lamoureux *et al*., 2002). Figure 2A shows a reasonable agreement between ΔΔ*G* predictions carried out with MatrixREDUCE and structural models. The close correspondence with the contact model, which is a function of the number of protein side chains in contact with DNA base pairs, is especially remarkable, showing that the MatrixREDUCE approach is capable of reproducing structural details of the binding interface based only on the genomic sequence and genome-wide TF occupancy data.

Gcn4 is a TF of the bZIP class. It is a homodimer with the basic region mediating sequence specific DNA binding and the leucine zipper region required for dimerization (O'Shea *et al*., 1991). For deriving the Gcn4 structural PSAMs, we used a 2.9 Å crystal structure of the TF bound to the ATGAGTCAT site (Ellenberger *et al*., 1992). The symmetry of the binding site (two reverse complement 4 bp half-sites separated by G in the middle) is a reflection of the homodimeric binding and is captured well in MatrixREDUCE predictions. While contact model and MatrixREDUCE predictions are similar, the all-atom model is less successful, probably due to the low resolution of the crystal structure, which leads to considerable uncertainty in side chain positions with respect to the neighboring DNA bases.

Finally, Rap1 binds DNA as a homodimer in a way that makes its DNA site a tandem repeat. The crystal structure of the Rap1 homodimer in complex with a telomeric DNA site has been solved to 2.25 Å resolution (Konig *et al*., 1996). Comparison of MatrixREDUCE PSAMs and structural PSAMs reveals good agreement with the all atom model. The contact model overpredicts binding specificity at the intermediate positions in the binding site (located

	Equal Probabilities		Intergenic Probabilities	
EMSA	R^2	P-value	R^2	P-value
Abf1 PBM	0.86	9.12×10^{-11}	0.85	2.92×10^{-10}
Abf1 ChIP YPD	0.87	6.63×10^{-11}	0.85	1.93×10^{-10}
Leu3 DIP 40nM	0.53	1.41×10^{-8}	0.53	1.59×10^{-8}
Leu3 DIP 4nM	0.42	1.40×10^{-6}	0.44	6.31×10^{-7}
Leu3 ChIP SM	0.45	5.08×10^{-7}	0.46	3.14×10^{-7}
Sum1 ChIP YPD	0.63	5.17×10^{-8}	0.67	7.50×10^{-9}
LacZ				
Abf1 PBM	0.78	6.39×10^{-8}	0.77	7.62×10^{-8}
Abf1 ChIP YPD	0.62	1.16×10^{-5}	0.63	8.99×10^{-6}
Sum1 ChIP YPD	0.41	3.89×10^{-6}	0.38	1.07×10^{-5}

Fig. 4. Correlations of experimentally measured $\Delta\Delta G$'s with information theory-predicted $\Delta\Delta G$'s. Experimental measurements of $\Delta\Delta G$'s were derived from EMSA and *lacZ* reporter assays. The R^2 and *P*-values for the Pearson correlations between the experimental and predicted $\Delta\Delta G$'s are presented for each PSSM-experimental data pair. PSSMs were derived and tested using two different background nucleotide frequencies: equal probabilities and intergenic probabilities.

between tandem repeats), likely because it assigns similar specificities to protein-DNA contacts in the loop region and in the DNA binding domains.

4.3 PSAM to PSAM correlations

Upon visual inspection of Figure 2A, the similarities are immediately apparent between affinity logos for the same factor inferred using different experimental and computational methods. However, a quantitative measure of these similarities can be obtained by aligning the PSAMs (see Methods) and calculating the correlation of their $\Delta\Delta G$ values. The *P*-value for this correlation between two PSAMs serves as our similarity metric (Figure 2B). Overall, the similarity between the PSAMs from MatrixREDUCE are the most significant. There is extreme similarity between the Rap1 PSAMs inferred from ChIP-chip and PBM data. The PSAMs inferred for Gcn4 for three different ChIP-chip conditions are all very similar as well. The Leu3 PSAMs inferred from the DIP-chip data are much more similar to each other than they are to the Leu3 PSAM inferred from the ChIP-chip data, but they are still both significantly similar (P-value < 0.01) to the ChIP-chip Leu3 PSAM. The significance of the correlations between MatrixREDUCE PSAMs and structurally inferred PSAMs is more variable. Both the all atom model PSAM and the contact model PSAM for Rap1 and Ndt80 have significant similarities with the respective MatrixREDUCE PSAMs (P-value < 0.01). However for Gcn4, while the contact model PSAM has strong similarities with all of the other PSAMs, the all atom PSAM has insignificant similarities with all other PSAMs.

4.4 How good is the information theory approximation?

In the original papers describing the ChIP-chip (Harbison *et al*., 2004), PBM (Mukherjee *et al*., 2004), and DIP-chip (Liu *et al*., 2005) data, the authors used BioProspector (Liu *et al*., 2001) or MDscan (Liu *et al*., 2002) to define weight matrix representations of TF binding sites. These two methods use the set of sequences that the experimenters label as ''bound'' to produce a list of potential binding sites. When interpreted through information theory, the nucleotide frequencies at each individual position in the binding sites divided by the ''background'' frequencies for the respective bases provide an estimate of a PSAM in the form of a position-specific scoring matrix (PSSM). Since we had already compiled the EMSA and *lacZ* expression data, we had the opportunity to experimentally verify the results of these PSSMs.

We gathered the BioProspector and MDscan results from the original, published analyses, transformed them into PSSMs, and used them to predict $\Delta\Delta G$'s for the EMSA and *lacZ* experimentally tested sequences. We performed this comparison using two different ''background'' nucleotide frequency models: one using equal nucleotide probabilities and one using nucleotide probabilities derived from *S. cerevisiae* intergenic sequences. The R^2 and *P*-values for the correlations between these predicted $\Delta\Delta G$'s and the experimental results are displayed in Figure 4. Overall, the quality of the results from the information theory PSSMs and the MatrixREDUCE PSAMs were similar. However, the results for the PSSMs are *different* depending on the choice of equal or intergenic nucleotide frequencies. While we did not test this scenario, the information theory results would also change depending on the probe intensity threshold chosen to label genes as ''bound.'' Thus, while MatrixREDUCE performs comparably with existing information theory methods, it conveniently avoids having to choose several *ad hoc* parameters required by the other methods.

5 DISCUSSION

Overall, position specific affinity matrices (PSAMs) as inferred by MatrixREDUCE from genome-wide TF occupancy data are good approximations of the real sequence-specific DNA binding affinities. Discrepancies between the computationally predicted and the experimentally inferred binding affinities may be due to either the computational or the experimental methods. EMSA has known problems with ''caging'' of the TFs by the gel while electrophoresis is proceeding (Fried and Crothers, 1981). This could lead to inferred $\Delta\Delta G$'s of smaller magnitude. Likewise, *lacZ* reporter assays are a very indirect way of measuring relative binding affinities as they require transcription, translation, and β-galactosidase reactions in order to make measurements, and noise could be introduced at each step. Structural model predictions are strongly dependent on the quality of input structures and are affected by errors in the energy function. The current MatrixREDUCE model may also give rise to systematic biases. First, it makes the approximation that nucleotides contribute independently to the free energy of TF binding (Benos *et al*., 2002). Second, it makes the assumption that the concentration of TF is much smaller than the K_d, which may not be correct for some TFs. Finally, all consecutive positions in the PSAM are currently treated as parameters to be estimated, which may lead to overfitting. We plan to address these issues in a future version of the algorithm.

Despite these current limitations, PSAMs discovered using the current implementation of MatrixREDUCE are good approximations of the relative nucleotide binding affinities of assayed TFs. Especially for microarray methods like PBM and DIP-chip, where the objective is to define nucleotide-binding specificities, MatrixREDUCE may be the most physically accurate method available to analyze the data. Even for less direct reflections of TF binding affinities like ChIP-chip or differential mRNA expression data, it will still provide good approximations of the sequence-specific binding affinities of TFs relevant to those data. Preliminary results also suggest that MatrixREDUCE performs well on data

from higher eukaryotes including *D. melanogaster* and mammals. Finally, MatrixREDUCE has two key advantages over most other computational methods for defining nucleotide binding specificities: (i) it uses the information for all probes in genome-wide TF occupancy data, and (ii) it does not require a background sequence model.

ACKNOWLEDGMENTS

This work was supported by National Institutes of Health grants HG003008 (to H.J.B) and CA121852. A.V.M. is a fellow of the Leukemia and Lymphoma Society. The authors thank Jason Lieb, Neil Clarke, Martha Bulyk, Ernest Fraenkel, Xiao Liu, and Michael Berger for graciously answering questions and supplying additional data from their publications. The authors also thank Gabor Halasz, Ronald Tepper, Junbai Wang, Xiang-Jun Lu, and Lucas Ward for valuable conversations, and Xiang-Jun Lu for reimplementing the MatrixREDUCE software for distribution.

REFERENCES

Benos,P.V., Bulyk,M.L. and Stormo,G.D. (2002) Additivity in protein-DNA interactions: how good an approximation is it? *Nucleic Acids Res.*, **30**, 4442–4451.

Berg,O.G. and von Hippel,P.H. (1987) Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. *J. Mol. Biol.*, **193**, 723–750.

Chu,S., DeRisi,J., Eisen,M., Mulholland,J., Botstein,D., Brown,P.O. and Herskowitz,I. (1998) The transcriptional program of sporulation in budding yeast. *Science*, **282**, 699–705.

Djordjevic,M., Sengupta,A.M. and Shraiman,B.I. (2003) A biophysical approach to transcription factor binding site discovery. *Genome Res.*, **13**, 2381–2390.

Djordjevic,M. and Sengupta,A.M. (2006) Quantitative modeling and data analysis of SELEX experiments. *Phys. Biol.*, **3**, 13–28.

Ellenberger,T.E., Brandl,C.J., Struhl,K. and Harrison,S.C. (1992) The GCN4 basic region leucine zipper binds DNA as a dimer of uninterrupted alpha helices: crystal structure of the protein-DNA complex. *Cell*, **71**, 1223–1237.

Endres,R.G., Schulthess,T.C. and Wingreen,N.S. (2004) Toward an atomistic model for predicting transcription-factor binding sites. *Proteins*, **57**, 262–268.

Foat,B.C., Houshmandi,S.S., Olivas,W.M. and Bussemaker,H.J. (2005) Profiling condition-specific, genome-wide regulation of mRNA stability in yeast. *Proc. Natl. Acad. Sci. USA*, **102**, 17675–17680.

Fried,M. and Crothers,D.M. (1981) Equilibria and kinetics of lac repressor-operator interactions by polyacrylamide gel electrophoresis. *Nucleic Acids Res.*, **9**, 6505–6525.

Gailus-Durner,V., Xie,J., Chintamaneni,C. and Vershon,A.K. (1996) Participation of the yeast activator Abf1 in meiosis-specific expression of the HOP1 gene. *Mol. Cell. Biol.*, **16**, 2777–2786.

Granek,J.A. and Clarke,N.D. (2005) Explicit equilibrium modeling of transcription-factor binding and gene regulation. *Genome Biol.*, **6**, R87.

Grubbs,F. (1969) Procedures for detecting outlying observations in samples. *Technometrics,* **11**, 1–21.

Harbison,C.T., Gordon,D.B., Lee,T.I., Rinaldi,N.J., Macisaac,K.D., Danford,T.W., Hannett,N.M., Tagne,J.B., Reynolds,D.B., Yoo,J., Jennings,E.G., Zeitlinger,J., Pokholok,D.K., Kellis,M., Rolfe,P.A., Takusagawa,K.T., Lander,E.S., Gifford,D.K., Fraenkel,E. and Young,R.A. (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**, 99–104.

Havranek,J.J., Duarte,C.M. and Baker,D. (2004) A simple physical model for the prediction and design of protein-DNA interactions. *J. Mol. Biol.,* **344**,59–70.

Issel-Tarver,L., Christie,K.R., Dolinski,K., Andrada,R., Balakrishnan,R., Ball,C.A., Binkley,G., Dong,S., Dwight,S.S., Fisk,D.G., Harris,M., Schroeder,M., Sethuraman,A., Tse,K., Weng,S., Botstein,D. and Cherry,J.M. (2002) Saccharomyces Genome Database. *Methods Enzymol.*, **350**, 329–346.

Iyer,V.R., Horak,C.E., Scafe,C.S., Botstein,D., Snyder,M. and Brown,P.O. (2001) Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, **409**, 533–538.

Konig,P., Giraldo,R., Chapman,L. and Rhodes,D. (1996) The crystal structure of the DNA-binding domain of yeast RAP1 in complex with telomeric DNA. *Cell,* **85**, 125–136.

Kortemme,T., Morozov,A.V. and Baker,D. (2003) An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein-protein complexes. *J. Mol. Biol.*, **326**, 1239–1259.

Lamoureux,J.S., Stuart,D., Tsang,R., Wu,C. and Glover,J.N.M. (2002) Structure of the sporulation-specific transcription factor Ndt80 bound to DNA. *EMBO J.*, **21**, 5721–5732.

Lee,T.I., Rinaldi,N.J., Robert,F., Odom,D.T., Bar-Joseph,Z., Gerber,G.K., Hannett,N.M., Harbison,C.T., Thompson,C.M., Simon,I., Zeitlinger,J., Jennings,E.G., Murray,H.L., Gordon,D.B., Ren,B., Wyrick,J.J., Tagne,J.B., Volkert,T.L., Fraenkel,E., Gifford,D.K. and Young,R.A. (2002) Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, **298**, 799–804.

Lieb,J.D., Liu,X., Botstein,D. and Brown,P.O. (2001) Promoter-specific binding of Rap1 revealed by genome-wide maps of protein-DNA association. *Nat. Genet.*, **28**, 327–334.

Liu,X.S., Brutlag,D.L. and Liu,J.S. (2002) An algorithm for finding protein-dna binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nat. Biotechnol.*, **20**, 835–839.

Liu,X., Brutlag,D.L. and Liu,J.S. (2001) Bioprospector: discovering conserved dna motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.* 127–138.

Liu,X. and Clarke,N.D. (2002) Rationalization of gene regulation by a eukaryotic transcription factor: calculation of regulatory region occupancy from predicted binding affinities. *J. Mol. Biol.*, **323**, 1–8.

Liu,X., Noll,D.M., Lieb,J.D. and Clarke,N.D. (2005) DIP-chip: rapid and accurate determination of DNA-binding specificity. *Genome Res.*, **15**, 421–427.

Lu,X.J. and Olson,W.K. (2003) 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic Acids Res.*, **31**, 5108–5121.

Morozov,A.V., Havranek,J.J., Baker,D. and Siggia,E.D. (2005) Protein-DNA binding specificity predictions with structural models. *Nucleic Acids Res.,* **33**, 5781–5798.

Mukherjee,S., Berger,M.F., Jona,G., Wang,X.S., Muzzey,D., Snyder,M., Young,R.A. and Bulyk,M.L. (2004) Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. *Nat. Genet.*, **36**, 1331–1339.

Olson,W.K., Gorin,A.A., Lu,X.J., Hock,L.M. and Zhurkin,V.B. (1998) DNA sequence-dependent deformability deduced from protein-DNA crystal complexes. *Proc. Natl. Acad. Sci. USA*, **95**, 11163–11168.

Onufriev,A., Bashford,D. and Case,D.A. (2004) Exploring protein native states and large-scale conformational changes with a modified Generalized Born model. *Proteins*, **55**, 383–394.

O'Shea,E.K., Klemm,J.D., Kim,P.S. and Alber,T. (1991) X-ray structure of the GCN4 leucine zipper, a two-stranded, parallel coiled coil. *Science*, **254**, 539–544.

Paillard,G. and Lavery,R. (2004) Analyzing protein-DNA recognition mechanisms. *Structure*, **12**, 113– 122.

Pierce,M., Benjamin,K.R., Montano,S.P., Georgiadis,M.M., Winter,E. and Vershon,A.K. (2003) Sum1 and Ndt80 proteins compete for binding to middle sporulation element sequences that control meiotic gene expression. *Mol. Cell. Biol.*, **23**, 4814–4825.

Ren,B., Robert,F., Wyrick,J.J., Aparicio,O., Jennings,E.G., Simon,I., Zeitlinger,J., Schreiber,J., Hannett,N., Kanin,E., Volkert,T.L., Wilson,C.J., Bell,S.P. and Young,R.A. (2000) Genome-wide location and function of DNA binding proteins. *Science*, **290**, 2306–2309.

Schneider,T.D. and Stephens,R.M. (1990) Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res.*, **18**, 6097–6100.

Simon,I., Barnett,J., Hannett,N., Harbison,C.T., Rinaldi,N.J., Volkert,T.L., Wyrick,J.J., Zeitlinger,J., Gifford,D.K., Jaakkola,T.S. and Young,R.A. (2001) Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, **106**, 697–708.

Stormo,G.D. and Fields,D.S. (1998) Specificity, free energy and information content in protein-DNA interactions. *Trends Biochem. Sci.*, **23**, 109–113.

Stormo,G.D., Schneider,T.D. and Gold,L. (1986) Quantitative analysis of the relationship between nucleotide sequence and functional activity. *Nucleic Acids Res.*, **14**, 6661–6679.

Stormo,G.D. (2000) DNA binding sites: representation and discovery. *Bioinformatics*, **16**, 16–23.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e150–e157
doi:10.1093/bioinformatics/btl243

MotifCut: regulatory motifs finding with maximum density subgraphs

Eugene Fratkin[1,*], Brian T. Naughton[2,*], Douglas L. Brutlag[2] and Serafim Batzoglou[1]

[1]Department of Computer Science, Stanford University, California, 94305 USA and [2]Department of Biochemistry, Stanford University, California, 94305 USA

ABSTRACT

Motivation: DNA motif finding is one of the core problems in computational biology, for which several probabilistic and discrete approaches have been developed. Most existing methods formulate motif finding as an intractable optimization problem and rely either on expectation maximization (EM) or on local heuristic searches. Another challenge is the choice of motif model: simpler models such as the position-specific scoring matrix (PSSM) impose biologically unrealistic assumptions such as independence of the motif positions, while more involved models are harder to parametrize and learn.

Results: We present MotifCut, a graph-theoretic approach to motif finding leading to a convex optimization problem with a polynomial time solution. We build a graph where the vertices represent all k-mers in the input sequences, and edges represent pairwise k-mer similarity. In this graph, we search for a motif as the maximum density subgraph, which is a set of k-mers that exhibit a large number of pairwise similarities. Our formulation does not make strong assumptions regarding the structure of the motif and in practice both motifs that fit well the PSSM model, and those that exhibit strong dependencies between position pairs are found as dense subgraphs. We benchmark MotifCut on both synthetic and real yeast motifs, and find that it compares favorably to existing popular methods. The ability of MotifCut to detect motifs appears to scale well with increasing input size. Moreover, the motifs we discover are different from those discovered by the other methods.

Availability: MotifCut server and other materials can be found at motifcut.stanford.edu

Contact: fratkin@cs.stanford.edu

1 INTRODUCTION

The identification of over-represented but imperfectly conserved motifs in genomic DNA is a problem with important biological applications, such as the discovery of regulatory elements that determine the timing, location, and level of gene transcription. Experimental techniques such as ChIP-chip and gene-expression microarrays can identify sets of genomic regions that are likely to be enriched for binding sites of a given transcription factor, which can then be mined computationally for an associated binding motif. This problem has been tackled many times with a number of disparate methods (Bulyk *et al*., 2003; Buhler *et al*., 2002; Eskin *et al*., 2002; Favorov *et al*., 2004; Frith *et al*., 2004; Gordon *et al*., 2005; Hertz *et al*., 1999; Hughes *et al*., 2000; Liang *et al*., 2004; Keich *et al*., 2002; Liu *et al*., 2001; Mahony *et al*., 2005; Pavesi *et al*., 2004; Pevzner *et al*., 2000; Sinha *et al*., 2004, 2003; Stormo *et al*., 1989; Thijs *et al*., 2001; Van Helden *et al*., 1998; Wang *et al*., 2003; Workman *et al*., 2000). A thorough examination of the field has also been published (Buhler *et al*., 2005).

There are two broad classes of motif-finding methods: probabilistic, and discrete or word-based. The most popular probabilistic methods model motifs with position-specific scoring matrices (PSSM). CONSENSUS (Stormo *et al*., 1989) uses a greedy strategy to attempt to maximize the information content of a position-specific scoring matrix (PSSM). AlignACE (Hughes *et al*., 2000) and BioProspector (Liu *et al*., 2001) use a Gibbs sampling strategy to search the space of all possible PSSMs. MEME (Bailey *et al*., 1995) models motifs similarly, and searches for motifs with a strategy based on Expectation Maximization (EM). The second class of motif-finding methods is discrete, or word-based searches. There is a diverse array of such methods, including clustering methods, such as WINNOWER/cWINNOWER (Liang *et al*., 2004) and PROJECTION (Buhler *et al*., 2002), and word-enumeration methods such as MDScan (Liu *et al*., 2002) (a ChIP-chip specific motif finding algorithm), MULTIPROFILER (Keich *et al*., 2002), and WEEDER (Pavesi *et al*., 2004).

Most of the popular approaches have a built-in assumption that the probability of each nucleotide at each position in the motif is independent of the nucleotides at other positions. Recent work has shown evidence for dependencies between positions in transcription factor binding sites (Benos *et al*., 2002; Bulyk *et al*., 2002; Zhou *et al*., 2004). Zhou and Liu found evidence for statistically significant dependencies in 25% of TRANSFAC motifs (Zhou *et al*., 2004). Our analysis of yeast and multicellular eukaryotic motifs confirms this (results not shown). Figure 1 is an example of a regulatory motif with nucleotide dependencies that cannot be accurately described with a simple PSSM model. To better model motifs that do not follow the simple PSSM model, some algorithms apply a Bayesian network framework (Agarwal *et al*., 1998; Barash *et al*., 2003; Friedman, 2003), while another approach uses a simpler model of pairwise dependencies (Zhou *et al*., 2004). These approaches provide added expressive power, but due to training issues and computational complexity have not yet been widely used in real life applications. In this paper, we reexamine the motif-finding problem from a novel, graph-theoretic perspective, which addresses the problem of nucleotide dependencies in a natural way. We formulate motif finding as a search for the maximum density subgraph of a graph whose nodes are the words in the input

*These authors contributed equally.
To whom correspondence should be addressed.

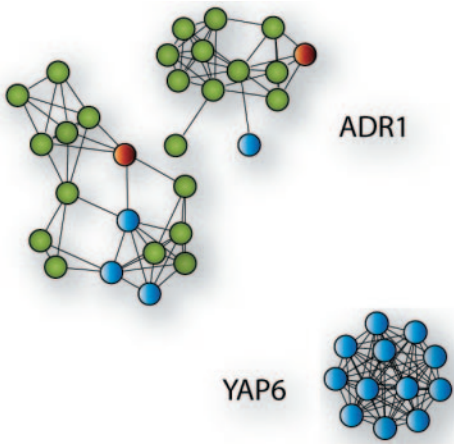


Fig. 1. Two yeast motifs: with and without nucleotide dependencies. This diagram shows two graphs of real yeast motifs—ADR1 and YAP6. Each node corresponds to a motif occurrence. Edges connect pairs of k-mers that are identical or differ by one mutation. If we model a motif with a PSSM, we can compute the probability of a specific k-mer being generated by that PSSM. Given the number of motif instances we can convert these probabilities into the expected number of occurrences for each k-mer. This number can be compared with the actual number of occurrences. In the two graphs k-mers that occur less frequently than expected are colored red, k-mers that occur more frequently are colored green, and cases in which observed and expected numbers are equal are colored blue. In such a graphical representation, PSSM-generated motifs have a single dense center, corresponding to the maximum likelihood k-mers, and the density of k-mers decreases as they are further from that center and hence less likely. The PSSM model is a good fit for the YAP6 motif, but not for the ADR1 motif.

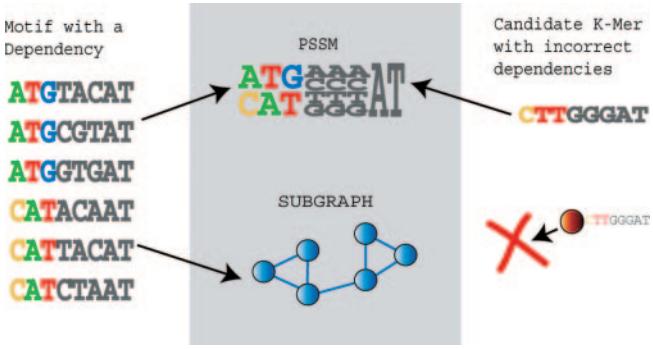


Fig. 2. Nucleotide dependencies and MDS. An alignment of 8-mers on the left represents a motif consisting of 6 motif occurrences. This is an artificial motif with perfectly conserved nucleotide dependencies (at positions 1, 2 and 3). The example k-mer on the right does not have the appropriate dependencies, and as such is not a good candidate motif. From the standpoint of the PSSM (center top), this k-mer appears to be as good a candidate as any of the motif instances. In contrast, if we create an edge between all the k-mers that have a mutation distance of 3 and fewer nucleotides and no edge otherwise, then the motif candidate will not be connected to the motif set in the graph representation. This example demonstrates that nucleotide dependencies that are ignored by the PSSM representation are implicitly incorporated in our graph representation of a motif.

sequences, and whose edges connect similar words. The resulting optimization can be performed in polynomial time. We present MotifCut, a tool for motif finding based on this formulation. Our method can be considered non-parametric, in the sense that the model size, in our case the number of vertices in the predicted motif subgraph, increases with the size of the input. MotifCut required minimal training and exhibits comparable running time to the leading motif finding algorithms. We tested MotfiCut on both simulated and real yeast data, and showed that it performs significantly better than previous leading approaches.

2 OVERVIEW OF THE MOTIFCUT ALGORITHM

As a first step, MotifCut converts the input sequences into a collection of k-mers. This collection contains all occurrences of k-mers in the sequences, where each overlap or duplicate is considered as a distinct k-mer. In other words, there is one k-mer for each nucleotide position in the input sequences. These k-mers form the set of vertices, V, in a graph $G = (V, E)$ representing the input. For every pair of vertices v_i, v_j we create an edge with a weight w_{ij}. The weight is a function of the number of mismatches between the two vertex k-mers, which is normalized with respect to the nucleotide background distribution, so that more similar k-mers are connected with higher-weight edges. The background distribution is used to find the probability of the two k-mers appearing at random given the input. Therefore, the weight of the edges connecting a pair of k-mers that are unlikely to appear in the background is up-weighted. Using these dependencies is essential, since it is well known that in genomic sequences certain dimers and trimers are much more common than the GC content alone would indicate (Karlin *et al*. (1997)).

A general motif can be viewed as a set of k-mers where elements of the set exhibit higher degree of pairwise similarity that would be expected based on the background distribution. Since an edge in this graph representation is a measure of that pairwise similarity, we expect the sets of k-mers representing motif occurrences to be recognizable as the portions of the graph with the greatest edge density. This insight is the basis of our algorithm. The notion of subgraph density can be defined in various ways, but there are only a few computationally tractable options. Among them is the most common definition of graph density: given a graph $G = (V, E)$, the density of G is $\|E\|/\|V\|$, where $\|V\|$ is the number of vertices and $\|E\|$ is the total weight of all the edges. We build on this formulation and define motif finding in the input sequences as the search for the *maximum density subgraph* (MDS) $G^* = argmax_{G' \subseteq G}(\|E'\|/\|V'\|)$. In *Methods* we will provide some evidence that this choice is reasonable for biological data. To search efficiently for maximum density subgraphs in genomic sequences, we introduce optimizations that we discuss in *Methods*.

The MDS formalization has two main advantages over PSSM-based methods, and most existing methods in general: (1) The motif model can in principle accommodate complicated and hard-to-parametrize structures in real motifs, such as nucleotide dependencies. Figure 2 illustrates how nucleotide dependencies can lead to a k-mer being incorrectly identified as a motif occurrence under the PSSM definition, and how the MotifCut algorithm deals with this problem. (2) Through problem-specific optimizations, motifs can be efficiently located in large inputs; the optimization problem that we define can be explicitly solved in polynomial time, and therefore the algorithm is guaranteed not to be trapped in local optima as input size increases.

3 RESULTS

3.1 Synthetic data

To test the performance of MotifCut we ran a series of tests against three of the most popular motif-finding algorithms currently available: MEME (Bailey *et al*., 1995), AlignAce (Hughes *et al*., 2000), and BioProspector (Liu *et al*., 2001). The selection of these three algorithms was largely based on an extensive performance analysis (Buhler *et al*., 2005). Of the 14 algorithms presented in that study, we considered the six that had the best performance in yeast and overall. From these six we chose three that did not have k-mer size limitations that would prevent them from discovering some of the real yeast motifs. The WEEDER (Pavesi *et al*., 2004) tool, for instance, does not operate on motifs of odd length or on k-mer sizes greater then 14; this would prevent it from being used on over 65% of real yeast data. The three algorithms that do not suffer from such limitations included both MEME and MEME3, and we chose to include only MEME since it demonstrated significantly better performance on yeast. We included BioProspector even though it was absent from the performance comparison (Liu *et al*., 2001), because in our tests it demonstrated the best performance of all previous methods.

Our first experiment consisted of running all of the algorithms on a synthetic data set. There are several reasons to use synthetic data for benchmarking. It obviates the problem of obtaining sufficient test cases, and allows us to gauge performance as a function of specific input parameters. It further eliminates the possibility that algorithm parameters were overtrained on known yeast motif annotations. Also, one can identify false positives unambiguously, while in real data some of the true motifs may not have been annotated.

For the synthetic data, we generated background sequences using a 3^{rd}-order Markov model. These dependencies were estimated from all intergenic regions of the yeast genome. We then generated three sets of PSSMs of sizes 8, 12, and 16, with fixed information contents of 12, 14, and 16 bits respectively (in a PSSM representation of a motif each nucleotide position can be viewed as 2 bits of information if it predicts a specific nucleotide with probability 1; if any of the four values are equally likely the information content at that position is 0). For each motif size, we run experiments on inputs consisting of 10,000, 15,000 and 20,000 nucleotides respectively, subdivided into 20 sequences. For each test, 20 instances of the motif were randomly seeded into the input, not necessarily one per sequence, and our results are based on 100 runs of each of the nine tests. To score the results we compute the PSSM of the top three motifs returned by each algorithm and calculate the value of its Pearson correlation with the seeded motif's PSSM. If the value of this correlation is higher than 0.7, which is a commonly used threshold for strong statistical similarity, we consider the seeded motif to be correctly identified. It is possible that an algorithm will find the true motif, but that its positions will be shifted left or right. For scoring purposes, we allow uniform shifts of one to three nucleotides across the entire motif set. The size of the maximum allowable shift depends on the k-mer size.

As can be seen in Figure 3, MotifCut performs better than the other methods in all of the tests on randomly generated data. All algorithms experience a significant performance drop-off as the ratio of the number of motif occurrences to the length of the input sequence decreases. MotifCut also follows this pattern, yet demonstrates a more gradual performance decay.

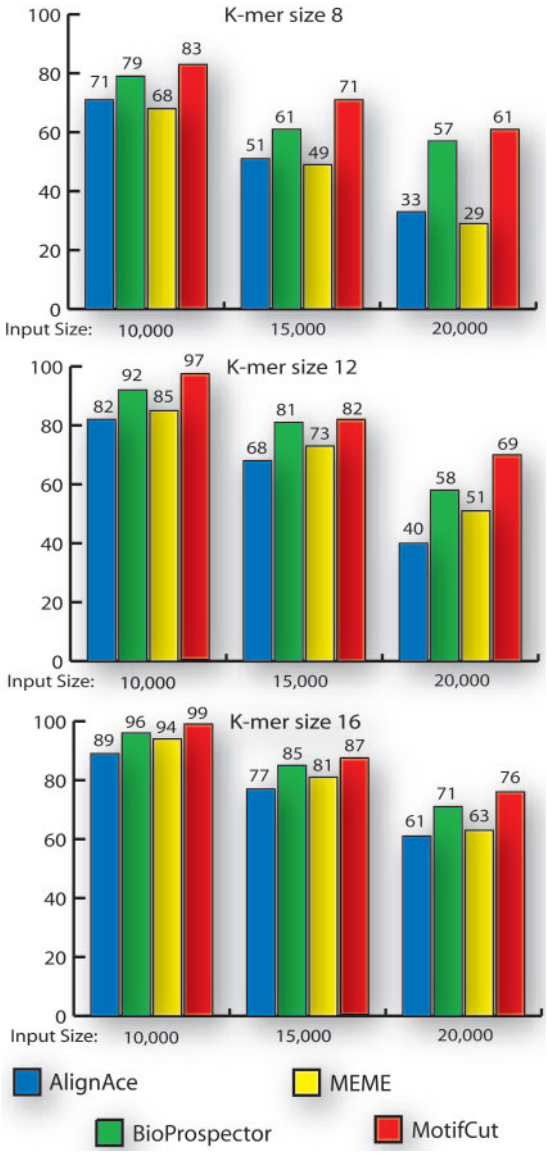


Fig. 3. Performance of four motif-finding algorithms on synthetic data. In these graphs the X-axis represents the input size, in nucleotides, and the Y-axis represents the percentage of motifs correctly identified. A motif is considered correctly identified if its Pearson correlation with the seeded motif is 0.7 or greater.

Since we use a novel formulation of the motif finding problem, we expected a lower correlation between the motifs found by MotifCut, and those located by the other three algorithms. One way to measure the correlation between two algorithms' performance is by using a log-odds ratio. Based on the test runs, we first calculate the probability of a motif being correctly identified simultaneously by both algorithms. We divide this probability by the expected probability under the assumption of algorithm independence, which is derived from the fraction of motifs found by each algorithm. Taking the log_2 of that ratio as the measure of similarity between two algorithms, a value of 0 indicates total independence, and a value of 1 signifies that the amount of observed

Table 1. Table of log-odds ratios on synthetic data

	AlignAce	BioProspector	MEME
MotifCut	0.14	0.10	0.12
MEME	0.20	0.31	
BioProspector	0.24		

The log-odds ratios of the 4 algorithms in the synthetic data set.

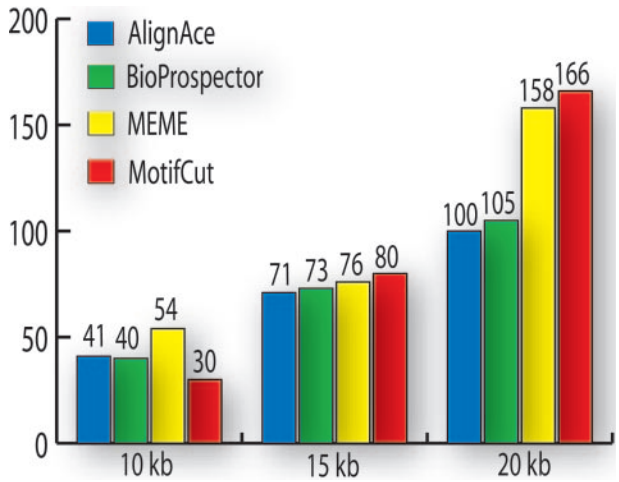


Fig. 4. Running time comparison. The X-axis is the input size, the Y-axis is the time in seconds in which the average task completes. The values are found by averaging running time on k-mers of sizes 8, 12 and 16.

overlap is twice the expected overlap assuming independence. Under this measure, unlike with the Pearson correlation, two strongly performing algorithms will not *a priori* have high similarity. Some motifs may be extremely easy to locate, whereas others may not be identifiable by any statistical methods. The correlation of all the algorithms in such instances is misleading since it is virtually methodology independent. Hence in computating the log-odds ratio we restricted our input to motifs that were found by at least one algorithm, and missed by at least one other algorithm. Table 1 shows the log-odds ratios for each pair of algorithms. As can be seen from the table, MotifCut's results are significantly different from those of the other three algorithms.

In our approach to motif finding we have departed from linear complexity sampling heuristics. To make the running time of MotifCut scale for real life applications we implemented a series of problem-specific optimizations, which made the MDS algorithm sub-quadratic. Details of our optimizations are discussed in Methods. Figure 4 demonstrates the running times for all four algorithms on synthetic test data. For each algorithm we benchmark the running time for input sequences of length 10,000, 15,000 and 20,000 base pairs. For each type we averaged running time for k-mers of size 8, 12 or 16. As can be seen from the figures, MotifCut's running time is comparable to that of the other algorithms.

3.2 Yeast data

Synthetic data is a convenient testbed that gives us control over every aspect of the input; however, it exhibits only limited fidelity to the real task. To benchmark the performance of our algorithm on real data, we tested MotifCut against other algorithms on a set of 83 experimentally verified yeast motifs, which were obtained in a genome-wide study that was reported previously (Harbison *et al.* (2004)). These motifs were identified in a CHip-chip experiment as having a p-value of less then 0.001 and were conserved in at least one other yeast genome.

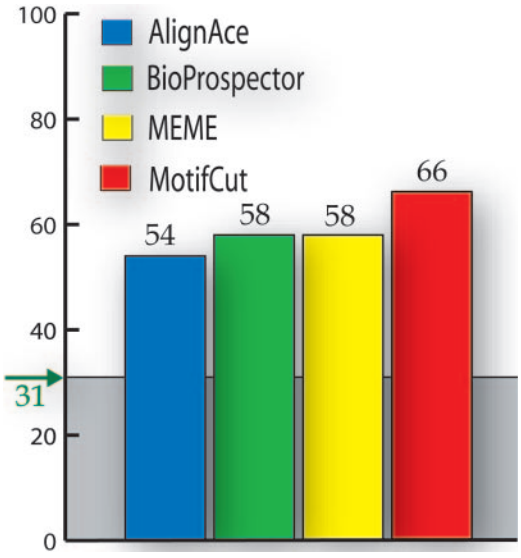


Fig. 5. Performance of the four algorithms on real yeast data. The Y-axis in this graph represents the percentage of motifs identified with a Pearson correlation of 0.7 or greater to the annotated motif's PSSM. The grey level represents the percentage of motifs found by all four algorithms.

Table 2. Table of correlations on yeast data

	AlignAce	BioProspector	MEME
MotifCut	0.13	0.12	0.02
MEME	0.67	0.38	
BioProspector	0.54		

This table shows log-odds correlation values between motifs found by each of the four algorithms in real yeast data.

As shown in Figure 5, MotifCut has a significant lead over the other methods in identifying yeast motifs. As was the case with the synthetic data, we accepted the top 3 results for each algorithm. 31% of the motifs were identified by all four algorithms.

Similarly to the synthetic data, we find that MotifCut identifies motifs that are less correlated with those found by other algorithms as shown in Table 2.

4 METHODS

4.1 Graph construction

The main idea behind the *MotifCut* algorithm is to formulate motif finding as the problem of finding the Maximum Density Subgraphs (MDS) in a specially constructed weighted graph $G = (V, E)$; where the set of vertices V corresponds to the set of all of the k-mer occurrences in the input, and the set of *undirected* edges E, represents nucleotide similarities between those

k-mers. Given such a graph G let $M \subseteq V$ denote the collection of k-mer occurrences corresponding to the binding sites of a specific transcription factor, and let B denote the background k-mers. The output of MotifCut it the set of vertices that is its best prediction of the set M.

Let $S_1 \ldots S_n$ be the set of input sequences. Each sequence S_i is an array of nucleotides $S_i = \{a_{i1}, a_{i2}, \ldots, a_{im}\}$. We start by constructing a collection of k-mers, V. This is the multiset of all k-mers encountered in the input including repeats and overlaps. Hence, each vertex v_l can be uniquely identified by its starting position. This position corresponds to a particular nucleotide a_{ij} in the input sequence, hence $v_l = [a_{ij}, a_{ij+1}, \ldots, a_{ij+k-1}]$ where k is the size of k-mers. The *edge set*, E, is created by a pairwise evaluation of the similarity between each pair of k-mers in V. Since G is a weighted graph, for simplicity we can view it as a fully connected graph and concentrate on assigning an appropriate weight to each edge.

The edge weight w_{ij} between a pair of vertices v_i and v_j is defined as follows:

$$w_{ij} = \frac{Pr(v_i \in M \mid v_j \in M) + Pr(v_j \in M \mid v_i \in M)}{\Theta(Pr(v_i \in B)) + \Theta(Pr(v_j \in B))} \quad (1)$$

In this formula Θ is a background normalization function that will be rigorously defined in *Background Normalization*. We estimate the other quantities as follows: To estimate the probability of $v_i \in M$ given $v_j \in M$ we took into account the k-mer size k and the *hamming distance* (number of mismatches) l between v_i and v_j. For every $k = 6, \ldots, 31$ we generated random PSSMs with a specific information content for every k, selected empirically to reflect the average conservation rates in yeast motifs (Harbison *et al*., 2004). Even though the PSSM model explicitly ignores inter-nucleotide dependencies, models effectively the majority of real motifs. On the other hand if we were to include data with simulated dependencies, the choices of those dependencies would reflect various examples found among annotated yeast motifs, and hence cause overfilling. For values of k where there are no curated databases of real motifs ($k > 18$), we extrapolated the information content based on the asymptotic behavior for $k = 6, \ldots, 18$. Then, we generated 100 instances of 10,000 bp-long input sequences by the 3^{rd}-order Markov background of the entire yeast genome; in each input, we implanted 20 occurrences of motifs generated by the PSSM. This resulted in 100 input graphs G, each containing 10,000 vertices, 20 of which are in M. Given $v_i \in M$, let $\alpha(k, l)$ be the number of vertices $v_j \in M$ that are l mismatches from v_i, and let $\beta(k, l)$ be the number of vertices $v_j \notin M$ that are l mismatches from v_i. For every combination of k and l we estimate

$$Pr(v_i \in M \mid v_j \in M) = \frac{\alpha(k,l)}{\beta(k,l)} \quad (2)$$

The plot of $Pr(v_i \in M \mid v_j \in M)$ against the number of nucleotide mismatches l between v_j and v_i is well approximated by a sigmoid. With parameters determined by the k-mer size k and the number of nucleotide mismatches between v_i and v_j.

$$Pr\,(v_i \in M \mid v_j \in M) = \frac{1}{1 + e^{-y+zl}} \quad (3)$$

Here y and z depend on the size of the k-mer, k. It can be seen that the results of this approximation (Figure 6) fall within the range observed in yeast data.

The probability of $v_i \in B$ is a straightforward application of the n^{th} order *Markov dependency* assumed for the input:

$$Pr\,(v_j) = \left(\prod_{a_{it} \in v_j} Pr\,(a_{it} \mid a_{it-1} \ldots a_{it-n}) \right) \Big/ .25 \quad (4)$$

The order n of the Markov dependency is based on the size of the input to ensure sufficient sampling.

4.2 Finding the maximum density subgraph

To find the maximum density subgraph we use a modified parametric flow algorithm (Gallo *et al*., 1989). This is an example of *fractional programming*

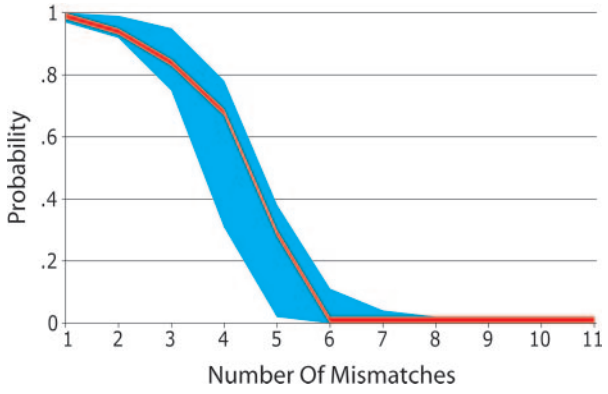


Fig. 6. The probability of a k-mer belonging to a motif given its mutation distance to a motif k-mer. The probability $Pr(v_i \in M \mid v_j \in M)$ is plotted (red line) as a function of number of mutations between v_j and v_i, as we estimated it from simulated PSSMs. The blue area is the range of values for the probability for motifs observed in yeast promoter regions. For this example we used k-mer size 11 since the amount of empirical data for this size was the greatest.

(Gallo *et al*., 1989). It is solved through an iterative application of the *push*/*relabel* algorithm to find *max-flow* and *min-cut*. To apply the parametric flow algorithm we modify our original graph G by adding two additional vertices: s—the source, and t—the sink. We compute the graph density $\lambda = \|E\|/\|V\|$. Each v_i of V will be connected to one of the vertices s or t. Let the sum of the weights of adjacent edges to v_i be $d(v_i)$. If $d(v_i) \geq \lambda$, s will be connected to v_i with an edge of weight $d(v_i) - \lambda$, otherwise v_i is connected to t with an edge of weight $\lambda - d(v_i)$. In the new graph we execute the *push*/*relabel* algorithm. This algorithm finds the *maximum flow* through the graph while also finding the *minimum cut*. The minimum cut partitions the graph into two disconnected subgraphs. One partition includes the sink and the other includes the source. We discard elements that are connected to the sink, recompute a new value for λ with the remaining elements and rerun the algorithm. This algorithm will converge in a polynomial number of steps to the set of vertices that constitute the maximum density subgraph. (Gallo *et al*., 1989).

The algorithm for finding the MDS is simple and easy to implement, however its running time is $O(nmlog(n^2m))$, where n is the number of vertices and m is the number of edges. This time complexity is too great for large data sets. To overcome this limitation, we define a set of n subgraphs $N(v_i)$, one for every vertex v_i. Each of these subgraphs represents a neighborhood of one of the vertices, v_i: it is induced by all vertices directly connected to v_i with an edge of weight greater than some threshold w, including v_i itself (Figure 7).

Since the graph G is fully connected, for a given motif set M there exists a minimal w such that for some instance of the motif, $\mu \in M$, all of the other instances of the motif are directly connected to μ with an edge weight greater than or equal to w. This means that all the instances of the motif will appear in a single neighborhood if the minimum allowed weight is less then or equal to w. Based on this property, for a proper choice of w, we can find the MDS of each local neighborhood, and one of these MDSs will contain the motif set. If we assume that the motif set M produces the highest density subgraph, then there will be a neighborhood of some motif instance μ that will produce the same density, and this density will be the highest of all the neighborhoods.

To speed up the algorithm, we want to set w as high as possible while not decreasing sensitivity of motif detection. We picked w using the following heuristic approach. To compute an appropriate threshold $w(k)$ as a function of motif length k, we generated motifs based on synthetic PSSMs of size k and with various information contents (bits of information), and implanted them in 10 kb of random sequence. We then found the information content

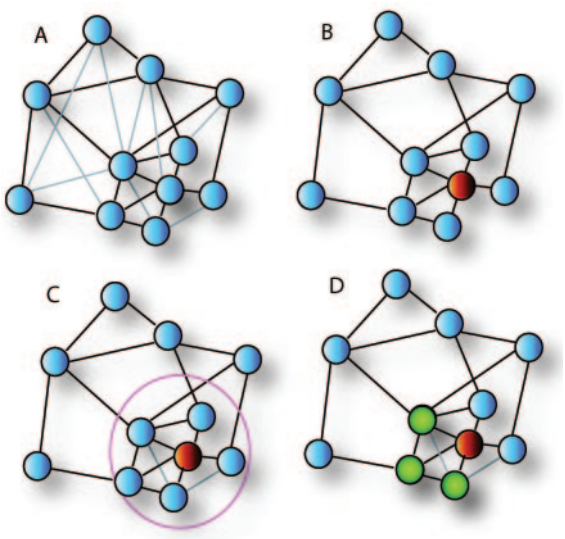


Fig. 7. Construction of a neighborhood subgraph. A. Starting with the complete graph, we first temporarily discard all the edges with weight $\leq w$ (depicted as light grey in the picture). B. Each vertex in turn is chosen (colored red), to be the center of a neighborhood. C. All the vertices connected to the central vertex define an induced subgraph; in this subgraph we reintroduce all the edges of lower weight than w. Those vertices and the central vertex form the neighborhood subgraph (circled with a red line). D. We find the maximum density subgraph of the neighborhood (green vertices). This process is repeated once for every vertex (k-mer) in the input.

such that 50% of motifs of that content were detected by running MDS on the entire graph G, and picked the highest $w(k)$ such that sensitivity did not decrease.

Therefore, after selecting all vertices that are connected to v_i with an edge weight > $w(k)$ (for k-mer size k), we obtain neighborhood graph $N(v_i)$. Note that this induced graph is defined to include *all* edges in E connecting pairs of nodes in the neighborhood of v_i, including edges of weight < $w(k)$. For each neighborhood subgraph $N(v_i)$ we find the maximum density subgraph $N'(v_i)$, and its associated density λ_i. Then, we isolate $N^* = argmax_{N'(v_i)}\lambda_i$. This latter subgraph is our candidate motif set that will be refined next.

4.3 The refinement step

Our set of candidate k-mer occurrences N^* is a good starting point for finding which elements constitute the real set of binding sites. However, empirically we often find false positives in this set. To eliminate false positives we find a subset of N^* that minimizes the entropy of the entire set. The entropy of a set S is computed as follows: $E(S) = log(\|S\|)\sum_{j=1}^{k}((\sum_{i\in S}a_{ji})\,log(\sum_{i\in S}a_{ji}))$. The optimization proceeds in a greedy leave-one-out fashion that finds the locally optimum subset. The process terminates once convergence is achieved. Note that the $log(\|S\|)$ term ensures that the set will not shrink to a single k-mer.

In practice we are often interested in finding more than a single motif candidate within a set of input sequences. We achieve that by performing the refinement step and returning a user-specified number of top scoring neighborhoods. We ensure that the returned results do not contain neighborhoods primarily consisting of the same vertices, or representing a uniform shift of a few bp over the vertex set of another neighborhood.

4.4 Background normalization

The function for assigning weights to the edges is critical for effectiveness of the method on real biological data. In particular real DNA often has

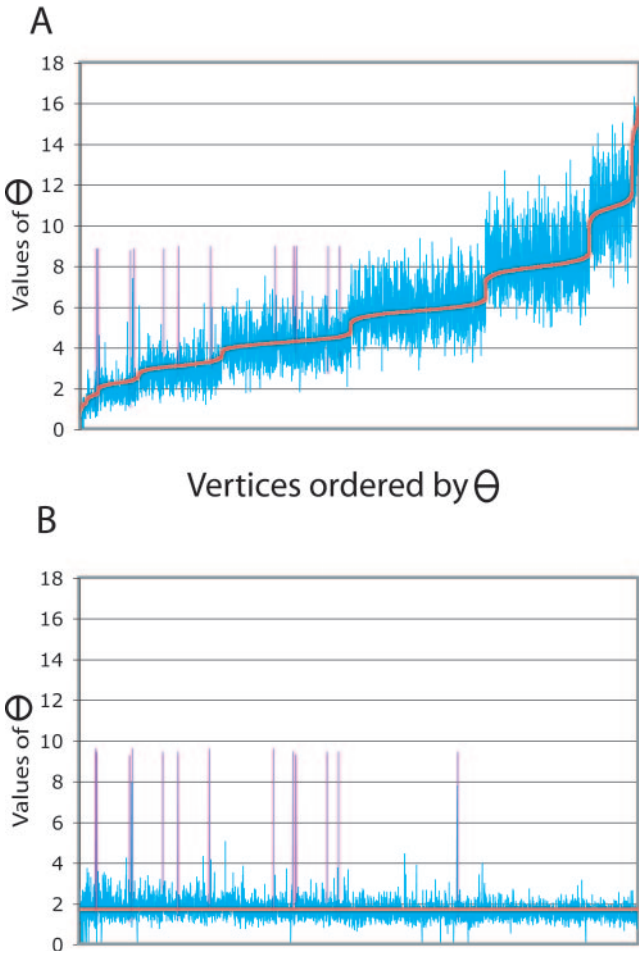


Fig. 8. Predicting k-mer specific density value. In graph A k-mers are sorted by the predicted maximum density (red line); the density that is calculated given each k-mer's nucleotide composition and its distribution in the background. The blue lines indicate actual maximum density values that were obtained from the neighborhood of the appropriate k-mer. Graph B shows the effect of normalization by the predicted maximum density. Once again, the red line demonstrates the normalized predicted value, and blue lines the actual value obtained. In both cases obvious outliers are instances of the motif.

irregularities in its k-mer distribution, such as a high *GC* content or low-complexity regions. If not accounted for, such abnormalities will emerge as dense regions in the graph, and obscure the presence of real motifs. We are compensating for these irregularities when computing edge weights, as was discussed in *Graph Construction*. Intuitively, this compensation should be such that in the absence of a true motif, for any vertices v_i, v_j, the corresponding densities λ_i and λ_j are approximately equal. To achieve that, we defined the function Θ in equation (1) empirically. We constructed input sequences with 3^{rd} order Markov irregularities of varying severity (up to 90% GC content). We then attempted different families of functions for Θ, such as exponential, polynomial, and logarithmic functions, and found that the following definition of Θ results in densities that are sufficiently normalized:

$$\Theta(Pr(v_j)) = Pr(v_j)^{1/e} \quad (5)$$

An example of how the normalization with Θ works in practice is displayed in Figure 8, for a set of sequences with 70% GC content. As seen in this figure, before normalization the densities λ_i display a high degree of vari-

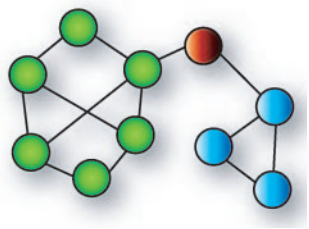


Fig. 9. The Maximum Density Subgraph for different exponent values. If we assume that all the edges in this graph are of weight 1, then the MDS for exponent 0.5 is of size 3 (blue vertices), and has a density $||E||^1/||V|| = 1$. The MDS for exponent 1, shown as green vertices, does not include the clique of size 3 and has a density of $||E||^1/||V|| = 4/3$.

ation, and as a result the motif instances are drowned by background words. After normalization, the background k-mers all induce approximately equal densities λ_i, and as a result the motif k-mers can be clearly discerned.

4.5 Choice of density function

As our results demonstrate, the objective function of subgraph density used in MotifCut is effective in separating motifs from background. However, it is not clear *a priori* that $||E||/||V||$ is the optimum choice of density function. Even though this function is a canonical definition of graph density, other functions are equally meaningful. A more general definition of density is given by $||E||^x/||V||$ where the exponent x can range from 0 to ∞. The value of x fundamentally changes the results of the optimization (Figure 9). If $x = 0.5$ for instance, and we are applying it to the unweighted graph, then this problem is equivalent to determining the size of the maximum clique. If $x = 2.0$ the output will always be the largest connected component of the graph. The optimization problem for most values of x is intractable. In fact, the only non-trivial formulation of the problem for which it remains tractable is for the exponent value $x = 1$, used by our algorithm. Though we cannot realistically use any other objective function than $||E||^1/||V||$, we also provide some evidence that our choice of x is likely to be close to optimal.

Ideally we would like to pick the exponent for which the densities of the motifs are separated as much as possible from maximum densities of subgraphs in the background. We asked the question of what is the best such exponent x for real yeast data. For each of the 83 promoter sets containing a motif, we first computed the density $||E||^x/||V||$ of the motif subgraph. Then, ideally we would like to compute the maximum density of a background subgraph. However, this is not possible for arbitrary x because the problem is intractable. Therefore, we relied on sampling random subgraphs. In every yeast input in our benchmark we picked 1,000 random subgraphs of size 2^y, where y ranged from 2 to 8, and measured their density for values of x from 0.5 to 2.0. For each combination of subgraph size and exponent x, we found the mean and standard deviation of the density values. The resulting distribution served as our estimate for the distribution of densities of background subgraphs. We now hypothesized that for a given subgraph size 2^y, the optimal choice of x is the one in which the ratio $R(x, y)$ of motif density over the standard deviation of the background density is maximum. For each exponent x we recorded the minimum ratio value produced by the background subgraphs of size 2^y (Figure 10A). When plotted against x, the minimum value $min_y R(x, y)$ creates a curve (Figure 10A, red line). The optimal choice for the exponent x on this curve corresponds to the highest point. In Figure 10B we plot the average over all yeast motifs of $min_y R(x, y)$, after we normalize that curve so that its peak is at 1. The peak should intuitively correspond to the optimal value of the exponent x, which we find to be 0.95. This value is remarkably close to the the canonical value of 1 used in MotifCut. Although the above argument is not a precise estimation of the objective function for our yeast data set, it nonetheless provides some explanation of the strong performance of MotifCut.

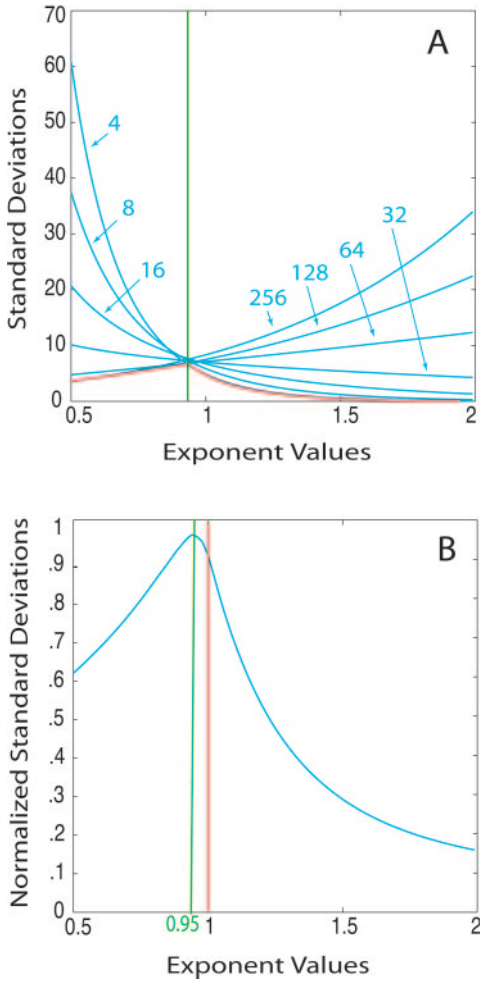


Fig. 10. Optimal exponent value. Graph A depicts an example of sampling output for a particular input (ADR1). X-axes are exponent values ranging from 0.5 to 2.0. The Y axis is the number of standard deviation for subgraphs of a specific size to the subgraph generated by the motif occurrences. Each blue line represents sample output for a particular size subgraph. Sizes range from 4 to 256. The red line indicates the minimum number of standard deviations for the specific value of the exponent. Graph B depicts the minimum values for each exponent, averaged over all yeast motifs. In image B the red line represents exponent used in MotifCut and green line represents the optimal value.

5 CONCLUSION

We have presented MotifCut, a novel graph-based approach to motif finding. We have demonstrated better performance than the leading motif finding algorithms on both simulated motifs, and experimentally derived yeast motifs. Performance of MotifCut appears to scale well with input size. An important feature of MotifCut is that the formulation is markedly different from the commonly used PSSM models, and as a result the motifs it finds are significantly different. Since originally the computational tools used to detect motifs in our yeast data set were PSSM (or ''diffused consensus'') based, it is possible that by imposing too many assumptions on the motif structure, motifs were missed. There is substantial evidence that cis-regulatory elements can evolve in parallel with

their binding factors (Athanikar *et al*., 1998; Jyoti *et al*., 1998; Shaw *et al*., 2002). For example, we see this effect in the transcription factor Rpn4p and its binding site in S. cerevisiae and C. albicans (Gasch *et al*., 2004). The sequences of these related binding sites can be substantially different. In our graph-based formulation, it is possible for two k-mers in a subgraph to be substantially different, if the k-mers that connect them are sufficiently edge-dense. Therefore, two related but substantially different k-mers can be part of the same motif, if there also exist a set of intermediary k-mers.

The source code and executables for MotifCut are available under the GNU public licence at http://motifcut.stanford.edu.

ACKNOWLEDGEMENTS

Work in the Batzoglou laboratory is supported in part by NSF grant EF-0312459, NIH grant U01-HG003162, the NSF CAREER Award, and the Alfred P. Sloan Fellowship. Brian Naughton is supported by the Stanford Biochemistry Department.

REFERENCES

Agarwal,P. and Bafna,V. (1998) Detecting non-adjacent correlations within signals in DNA. *RECOMB*, 2–8.

Athanikar,J.N. and Osborne,T.F. (1998) Specificity in cholesterol regulation of gene expression by coevolution of sterol regulatory DNA element and its binding protein. *Proc. Natl. Acad. Sci. U S A*, **95**(9), 4935–4940.

Bailey,T.L. and Elkan,C. (1995) The value of prior knowledge in discovering motifs with MEME. *ISMB*, 21–29.

Barash,Y., Elidan,G., Friedman,N. and Kaplan,T. (2003) Modeling dependencies in protein-DNA binding sites. *RECOMB*, 28–37.

Benos,P.V., Lapedes,A.S. and Stormo,G.D. (2002) Probabilistic code for DNA recognition by proteins of the EGR family. *J. Mol. Biol.*, **323**, 701–727.

Buhler,J., Tompa *et al*. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, **23**, 137–144.

Buhler,J. and Tompa,M. (2002) Finding motifs using random projections. *J. Comput. Biol.*, **9**(2), 225–242.

Bulyk,M.L. (2003) Computational prediction of transcription-factor binding site locations. *Genome Biol.*, **5**(1), 201.

Bulyk,M.L., Johnson,P.L.F. and Church,G.M. (2002) Nucleotides of transcription factor binding sites exert interdependent effects on binding affinities of transcription factors. *Nucleic Acid Research*, **30**(5), 1255–1261.

Eskin,E. and Pevzner,P. (2002) Finding composite regulatory patterns in DNA sequences. *Bioinformatics, Supplement 1*, **18**, 354–363.

Favorov,A.V., Gelfand,M.S., Gerasimova,A.V., Mironov,A.A. and Makeev,V.J. (2004) Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length and its validation on the ArcA binding sites. *BGRS*.

Friedman,N. (2003) Probabilistic models for identifying regulation networks. *Bioinformatics*, **57**.

Frith,M.C., Hansen,U., Spouge,J.L. and Weng,Z. (2004) Finding functional sequence elements by multiple local alignment. *Nucleic Acids Res.*, **32**, 189–200.

Gallo,G., Grigoriadis,M. and Tarjan,R. (1989) A fast parametric maximum flow algorithm and applications. *SIAM Computer Science*, **18**, 30–55.

Gasch,A.P., Moses,A.M., Chiang,D.Y., Fraser,H.B., Berardini,M. and Eisen,M.B. (2004) Conservation and evolution of cis-regulatory systems in ascomycete fungi. *PLoS Biol.*, **11**, 398.

Gordon,D.B., Nekludova,L., McCallum,S. and Fraenkel,E. (2005) TAMO: a flexible, object-oriented frame-work for analyzing transcriptional regulation using DNA-sequence motifs. *Bioinformatics*, **21**(14), 3164–3165.

Harbison,C.T. *et al*. (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**, 99–104.

Hertz,G.Z. and Stormo,G.D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**, 563–577.

Hughes,J.D., Estep,P.W., Tavazoie,S. and Church,G.M. (2000) Computational identification of cis-regulatory elements associated with functionally coherent groups of genes in Saccharomyces cerevisiae. *J. Mol. Biol.*, **296**, 1205–1214.

Jyoti,N., Osborne,A. and Osborne,F.T. (1998) Specificity in cholesterol regulation of gene expression by coevolution of sterol regulatory DNA element and its binding protein. *Biochemistry*, **95**(9), 4935–4940.

Karlin,S. and Mrazek,J. (1997) Compositional differences within and between eukaryotic genomes. *Proc. Natl. Acad. Sci. USA*, **94**(19), 10227–10232.

Keich,U. and Pevzner,P.A. (2002) Finding motifs in the twilight zone. *Bioinformatics*, **18**(10), 1374–1381.

Liang,S., Samanta,M.P. and Biegel,B.A. (2004) cWINNOWER algorithm for finding fuzzy dna motifs. *J Bioinform Comput Biol.*, **2**(1), 47–60.

Liu,X.S., Brutlag,D.L. and Liu,J.S. (2002) An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nat. Biotechnol.*, **20**(8), 835–839.

Liu,X., Brutlag,D.L. and Liu,J.S. (2001) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.*, 127–138.

Mahony,S., Hendrix,D., Golden,A., Smith,T.J. and Rokhsar,D.S. (2005) Transcription factor binding site identification using the self-organizing map. *Bioinformatics*, **1**(21), 1807–1814.

Pavesi,G., Mereghetti,P., Mauri,G. and Pesole,G. (2004) Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Res.*, **32**, 199–203.

Pevzner,P. and Sze,S.H. (2000) Combinatorial approaches to finding subtle signals in DNA sequences. *ISMB*, 269–278.

Sinha,S., Blanchette,M. and Tompa,M. (2004) PhyME: A probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics*, **5**, *170*.

Sinha,S. and Tompa,M. (2003) YMF: a program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res.*, **31**, 3586–3588.

Shaw,P.J., Wratten,N.S., McGregor,P.A., Dover,A.G. and McGregor,P.A. (2002) Coevolution in bicoid-dependent promoters and the inception of regulatory incompatibilities among species of higher Diptera. *Evolution and Development*, **4**(4), 265–277.

Stormo,G.D. and Hartzell,G.W. (1989) Identifying protein-binding sites from unaligned DNA fragments. *Proc Natl Acad Sci USA*, **86**(4), 1183–1187.

Thijs,G. *et al*. (2001) A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics*, **17**, 1113–1122.

Van Helden,J., Andre,B. and Collado-Vides,J. (1998) Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.*, **281**, 827–842.

Wang,T. and Stormo,G.D. (2003) Combining phylogenetic data with co-regulated genes to identify regulatory motifs. *Bioinformatics*, **19**(18), 2369–2380.

Workman,C.T. and Stormo,G.D. (2000) ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. *Pacific Symposium on Biocomputing*, 467–478.

Zhou,Q. and Liu,J.S. (2004) Modeling within-motif dependence for transcription factor binding site predictions. *Bioinformatics*, **4**, 909–916.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e158–e165
doi:10.1093/bioinformatics/btl201

Bistable switching and excitable behaviour in the activation of Src at mitosis

Hendrik Fuß*, Werner Dubitzky, Stephen Downes and Mary Jo Kurth

University of Ulster, School of Biomedical Sciences, Cromore Road, Coleraine, BT52 1SA, Northern Ireland

ABSTRACT

Motivation: The protein tyrosine kinase Src is involved in a multitude of biochemical pathways and cellular functions. A complex network of interactions with other kinases and phosphatases obscures its precise mode of operation.

Results: We have constructed a semi-quantitative computational dynamic systems model of the activation of Src at mitosis based on protein interactions described in the literature. Through numerical simulation and bifurcation analysis we show that Src regulation involves a bistable switch, a pattern increasingly recognised as essential to biochemical signalling. The switch is operated by the tyrosine kinase CSK, which itself is involved in a negative feedback loop with Src. Negative feedback generates an excitable system, which produces transient activation of Src. One of the system parameters, which is linked to the cyclin dependent kinase cdc2, controls excitability via a second bistable switch. This topology allows for differentiated responses to a multitude of signals. The model offers explanations for the existence of the positive and negative feedback loops involving protein tyrosine phosphatase alpha (PTPα) and translocation of CSK and predicts a specific relationship between Src phosphorylation and activity.

Contact: fuss-h@ulster.ac.uk

INTRODUCTION

One striking feature of the protein tyrosine kinase pp60/c-Src is the number of biochemical signalling pathways it is involved in. Cellular functions such as cytoskeletal organisation, cell cycle control, growth factor signalling, cell adhesion, migration and differentiation depend on it (Thomas and Brugge, 1997; Bjorge *et al*., 2000; Brown and Cooper, 1996). The numerous phosphorylation targets that have been identified characterise Src as a hub in cellular signalling networks (Abram and Courtneidge, 2000).

It was originally discovered as a viral oncogene (*v-Src*) (Rous, 1911) and in fact it is frequently deregulated in various human cancers (Irby *et al*., 1999; Thomas and Brugge, 1997). Its normal, physiological form, *c-Src*, occurs in all higher vertebrates. It also constitutes a family of tyrosine kinases consisting of nine members. Src homology domains (SH domains) appear as interaction domains in many other proteins. For example, SH2 is an important phosphotyrosine (pTyr) binding motif. Crystal structures of Src family proteins have provided many clues as to its regulation (Xu *et al*., 1997). However, despite its obvious importance to the correct functioning of cellular processes, a tangible role for Src in cell physiology has not yet emerged.

Thomas and Brugge identified six major difficulties that impede the functional characterisation of individual Src family kinases (Thomas and Brugge, 1997). Among these are: redundancy among Src family members, the complexity of their downstream pathways, interference due to a high degree of homology among interacting proteins and a few others that make the *in vivo* system inaccessible.

Our computer-aided approach aims to tackle this complexity through mathematical modelling and numerical analysis. We present here a dynamic systems model aiming to characterise the activation of Src at mitosis by simulating known and hypothesised interactions.

A dynamic systems model describes the temporal evolution of a system state using only a fixed set of rules. In a protein-protein interaction model the system state is defined by the concentrations of each molecular entity at a given time t. The dynamic rules of the model allow computation of a subsequent state at a later time $t + \Delta t$.

The objective of our approach is two-fold: First, complex biological systems require a reliable way of testing whether the hypothesised interactions account for the observed behaviour in the real system. The aim is to find reasons for the vast complexity of interactions that Src undergoes in the characteristics of its temporal behaviour. Second, dynamic models have predictive capabilities. A variety of experimental set-ups, such as overexpression and deletion or the effects of amino acid substitution, can be simulated with our model. Emergent properties such as bistability, robustness or sensitivity to system parameters, such as kinase activities or physiological conditions, can be verified in wet-lab experiments and thus provide new insights about the characteristics of the enzymes involved.

Current research is beginning to relate known protein-protein interactions to architectural concepts and principles (Tyson *et al*., 2003; Milo *et al*., 2002). One important concept is bistability, which is now recognised as an essential feature of cellular signalling networks (Bhalla and Iyengar, 1999). A bistable system can alternate between two discrete stable steady-states in response to a signal. Bistable biochemical switches can, for example, amplify and modulate an extra-cellular signal to yield a decisive all-or-nothing response from the intracellular logic. Enzymatic systems involved in cell cycle regulation appear to make heavy use of bistable systems in order to co-ordinate progression through each stage of the cycle (Fuß *et al*., 2005; Ingolia and Murray, 2004).

*To whom correspondence should be addressed.

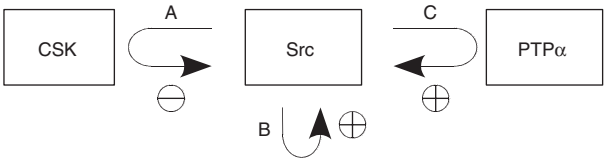


Fig. 1. Schematic representation of three feedback loops controlling the activity of Src. Two *external* feedback loops (A and C) involve other kinases (CSK) and phosphatases (PTPα), while *internal* feedback (B) is caused by autophosphorylation.

The Src system

As noted above, Src interacts with a huge number of different proteins. From these we have taken an exemplar set that is thought to be essential to the activation of Src at mitosis.

We know that Src activity is controlled in a complex manner through several phosphorylation sites: When phosphorylated on Tyr529[1] Src assumes a compact, inactive conformation due to an intramolecular pTyr-SH2 interaction (Xu *et al*., 1997). It is normally inactive in this conformation (with exceptions as we shall see later). The N-terminal sites Thr34, Thr46 and Ser72 are thought to be phosphorylated by the cyclin-dependent kinase cdc2 (Shenoy *et al*., 1989). Their phosphorylation leads to weakening of the intramolecular pTyr-SH2 interaction and exposes pTyr529 to phosphatases (Stover *et al*., 1994; Shenoy *et al*., 1992). Phosphorylation of the other major tyrosine phosphorylation site, Tyr418 within the activation loop, is known to significantly increase kinase activity (Kmiecik and Shalloway, 1987; Xu *et al*., 1999). This site is phosphorylated in an autocatalytic reaction.

Two other proteins are associated with phosphorylation and dephosphorylation of Tyr529: PTPα and CSK (see Fig. 1). PTPα (*Protein Tyrosine Phosphatase alpha*) is the ubiquitously expressed receptor protein tyrosine phosphatase (RPTP) that is known to dephosphorylate Src pTyr529 and thus to positively affect Src activity. Furthermore it has been suggested that PTPα is a target of Src itself: PTPα features at least one phosphorylation site at Tyr789 that is essential for its Src directed activity. It has been hypothesised that PTPα therefore participates in a positive feedback loop with Src (den Hertog *et al*., 1994; Zheng and Shalloway, 2001). The activity of PTPα is further regulated by an inhibitory protein, Grb2. Grb2 binding to PTPα down-regulates its activity (den Hertog *et al*., 1994; Zheng and Shalloway, 2001). However, serine hyperphosphorylation of PTPα can prevent their interaction. Zheng and Shalloway suggested that protein kinase C (PKC) could be responsible for this modification (den Hertog *et al*., 1994; Zheng and Shalloway, 2001).

CSK (*c-Src tyrosine kinase* or *C-terminal Src kinase*) is a negative regulator of Src activity and is homologous to Src. However, its activity is regulated in a different fashion. In the inactive state, CSK resides in the cytoplasm. Cbp (*CSK-binding protein*, also known as PAG), a membrane-located binding protein and phosphorylation target of Src, associates with CSK through a pTyr-SH2 interaction. This interaction enables its kinase activity and recruits it to the membrane, where it can target the inhibitory Src phosphorylation site Tyr529. Cbp and CSK therefore seem to constitute a negative feedback loop with Src (Howell and Cooper, 1994; Kawabuchi *et al*., 2000; Brdička *et al*., 2000).

[1]The nomenclature for Src phosphorylation sites is not consistent in the literature, which is partly due to the different positions of these sites in human, mouse and chicken homologues. All positions in this article refer to human Src according to UniProt entry P12931.

METHODS

Information retrieval

Our model is the result of a hypothesis-driven (as opposed to data-driven) approach. The information about protein interactions was mainly retrieved from published articles referenced throughout this paper.

Default parameter values were manually adjusted to fit observations from literature and to correspond to physiologically plausible assumptions. Where available, quantitative findings from literature have been accommodated. However, our model focuses on the discovery of general, qualitative behavioural features rather than on making quantitative predictions. The model variables therefore represent dimensionless concentrations.

Computational modelling

The software tool *Narrator*, an implementation of the *Codependence Model* formalism (Mandel *et al*., 2006), was used to define the Src system model in Fig. 2.

Codependence Models are a graphical formalism intended for describing dynamic models of complex biological systems. The main strength of the formalism is the uniform description of transport or (for example chemical) transformation processes and informational interactions, such as stimulation and inhibition (Mandel *et al*., 2006). The formalism also defines a dynamic interpretation and therefore allows direct translation to a system of ordinary differential equations (ODE). It is more readable and less prone to errors, since features like conservation of mass are implicitly derived from the graph structure. The equations that need to be specified (see Table 1) are much simpler than in a typical biochemical ODE model. Unlike many formalisms, its model elements are abstract entities rather than representations of concrete biochemical classes such as enzymes, proteins, transcription factors etc. Its one-to-one correspondence to ODEs enables the use of a full range of mathematical analysis techniques. The main elements of the Codependence Model formalism are described in the caption of Fig. 2.

Kinetic equations

Our model employs simple, mass-action based kinetics to represent enzymatic reactions. The resulting equations are simpler than for example Michaelis-Menten kinetics and contain fewer parameters. This simplification is valid under the assumption that the Michaelis constant K_M for each enzyme is large compared to its physiological concentration.

All kinetic equations (velocities) therefore take the form

$$v = \sum_i \nu_i k_i \cdot a_{\text{enzyme}} \cdot \text{substrate}$$

where ν_i is a stoichiometric parameter to distinguish forward and reverse reaction, k_i the kinetic rate constant and a_{enzyme} the activity of the relevant enzyme. Enzymatic activities of Src and PTPα also include a background element a^0_{enzyme}.

Numerical simulation

For subsequent mathematical analyses, the model was converted to an *ODE file* for use with XPPAut (Version 5.91) (Ermentrout, 2005) The latter is a numerical simulation program for ODEs and other types of equation systems. It includes an interface to AUTO (Doedel, 1981), a software package for numerical bifurcation analysis, which was used to generate Figures 3 to 7.

For numerical solution of differential equations a fourth order Runge-Kutta algorithm with adaptive step size was used, with a maximum step size of 0.1. AUTO was operated with an error tolerance of 10^{-7} (EPSL, EPSU and EPSS) and a step size not greater than 0.01.

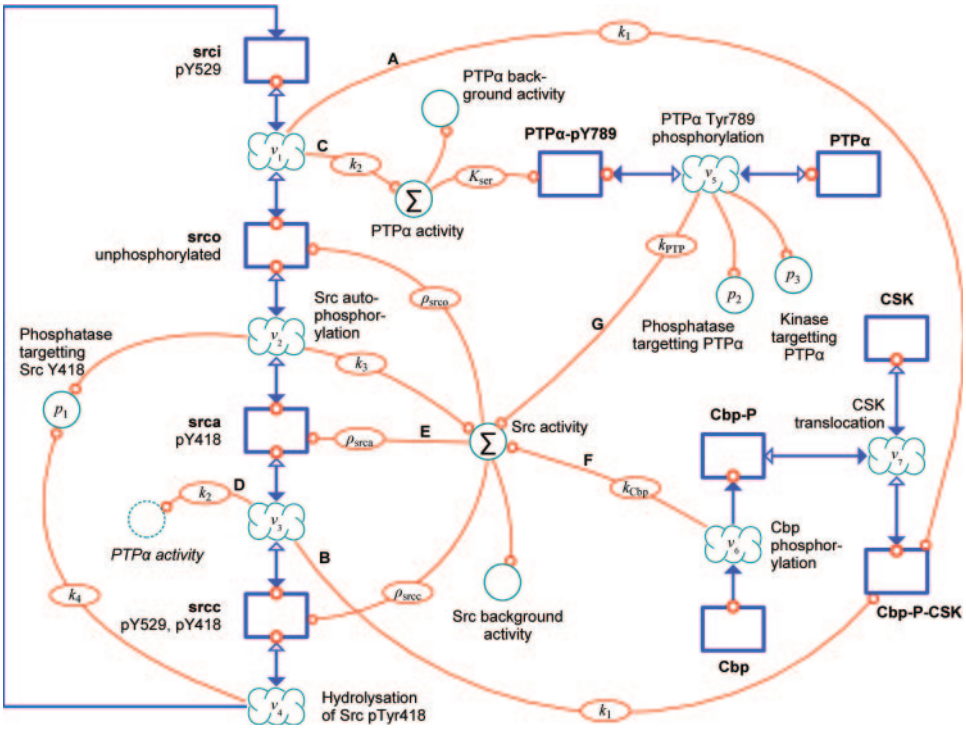


Fig. 2. Computational model. Blue rectangles represent molecular species, turquoise clouds denote reactions between them. Blue arrows represent material flow (positive direction is indicated by filled arrowhead). Thin red lines, so-called conditional links, represent informational flow, e.g. enzymatic activity. The small red circle indicates information source, e.g. the enzyme catalysing the reaction. Each condition has an associated weight that controls the intensity of the interaction (ellipses on conditional links). If unspecified, the weight is 1. Large turquoise circles represent constants or computed logical entities.

RESULTS

Model definition

We have constructed a dynamic model that represents interactions between Src, PTPα and CSK. These proteins were selected because of their known and described importance for tyrosine phosphorylation in the cell cycle, specifically at mitosis (Mustelin and Hunter, 2002). Zheng and Shalloway developed a basic temporal model for PTPα activation (Zheng and Shalloway, 2001, Fig. 10). Our approach integrates this model and the Cbp/CSK system (see for example Kawabuchi *et al*. (2000); Okada *et al*. (1991)) in a computational *dynamic systems model*.

A diagrammatic representation of the model, for which we have used a graphical formalism called Codependence Models (Mandel *et al*. (2006), see Methods), is shown in Fig. 2. The model consists of ten species, which hold the concentrations of each molecular entity. The model works with fractional units rather than physiological concentrations, since there is little quantitative data available. These entities are connected through seven reactions (process clouds in Fig. 2). Their velocities, ν_1 to ν_7, are defined using only basic mass-action kinetics (Table 1). Kinetic parameters are indicated as weights on the respective links (thin lines, known as conditional links).

Src occupies the central role in our model. Two tyrosine phosphorylation sites are of particular importance to the dynamics of Src activation, namely Tyr529 and Tyr418. These two sites allow four possible combinations of phosphorylation states, which are depicted on the left-hand side of Fig. 2. Other phosphorylation sites are not explicitly represented in the model, but the effect of N-terminal serine and threonine phosphorylation can be simulated through model parameters as we will see. The four states are abbreviated *srci* (inactive), *srco* (opened), *srca* (activated) and *srcc* (closed) for convenience. The corresponding states of each phosphorylation site can be seen in Fig. 2.

The model allows all conformations except *srci* to contribute to kinase activity, and each to a different degree, reflected by their specific molar activities ρ_{srco}, ρ_{srca} and ρ_{srcc}. The node '*Src activity*' combines these into a summative activity exerted onto Src phosphorylation targets, which includes Src itself (autophosphorylation, interaction E). Tyrosine kinase background activity is included as an additional model parameter.

Phosphorylation status of Tyr529 is controlled by CSK (interactions A and B in Fig. 2) and PTPα (interactions C and D). We assumed that the activity of these enzymes is independent of Tyr418 phosphorylation status, hence they control both ν_1 and ν_3.

While the *srcc* state (phosphorylated on both Tyr529 and Tyr418) has been observed *in vitro* (Boerner *et al*., 1996) its exact role *in vivo* remains controversial. The kinase domain of *srcc* appears to retain its activity. Once Src is activated, CSK is not able to deactivate it until pTyr418 is dephosphorylated (Sun *et al*., 1998). Xu *et al*. state that the *srcc* state has not been observed *in vivo* (Xu *et al*., 1999). It is unclear why CSK should not target *srca* in living cells. If *srcc* cannot be observed, this could be due to a short lifetime of this state. This is reflected in the model by a high default value of $k_4 = 10$, which leads to rapid hydrolysation of pTyr418 in *srcc*. We do not, however, propose or favour any particular mechanism of

Table 1. Mathematical definition of model components. v_1 to v_7 refer to processes in Fig. 2. Combined with the model topology these equations can be converted to a set of differential equations for numerical analysis.

Processes:

$$v_1 = k_2 \cdot a_{\mathrm{PTP}\alpha} \cdot \mathrm{srci} - k_1 \cdot \mathrm{cbp\text{-}P\text{-}csk} \cdot \mathrm{srco}$$
$$v_2 = k_3 \cdot a_{\mathrm{Src}} \cdot \mathrm{srco} - p_1 \cdot \mathrm{srca}$$
$$v_3 = k_1 \cdot \mathrm{cbp\text{-}P\text{-}csk} \cdot \mathrm{srca} - k_2 \cdot a_{\mathrm{PTP}\alpha} \cdot \mathrm{srcc}$$
$$v_4 = k_4 \cdot p_1 \cdot \mathrm{srcc}$$
$$v_5 = (a_{\mathrm{Src}} + p_3) \cdot \mathrm{PTP}\alpha - p_2 \cdot \mathrm{PTP}\alpha\text{-pTyr789}$$
$$v_6 = k_{\mathrm{cbp}} \cdot a_{\mathrm{Src}} \cdot \mathrm{cbp}$$
$$v_7 = k_{\mathrm{csk;on}} \cdot \mathrm{cbp\text{-}P} \cdot \mathrm{csk} - k_{\mathrm{csk;off}} \cdot \mathrm{cbp\text{-}P\text{-}csk}$$

Activities:

$$a_{\mathrm{Src}} = \rho_{\mathrm{srco}} \cdot \mathrm{srco} + \rho_{\mathrm{srca}} \cdot \mathrm{srca} + \rho_{\mathrm{srcc}} \cdot \mathrm{srcc} + a^0_{\mathrm{Src}}$$
$$a_{\mathrm{PTP}\alpha} = \mathrm{ptpy} + a^0_{\mathrm{PTP}\alpha}$$

Default parameters:

$$\rho_{\mathrm{srco}} = 0, \quad \rho_{\mathrm{srca}} = 1, \quad \rho_{\mathrm{srcc}} = 1$$
$$k_1 = 1.0, \quad k_2 = 0.8, \quad k_3 = 1.0, \quad k_4 = 10$$
$$k_{\mathrm{cbp}=1.0}, \quad k_{\mathrm{PTP}\alpha=1.0}, \quad K_{\mathrm{ser}} = 1,$$
$$k_{\mathrm{csk;on}} = 0.1, \quad k_{\mathrm{csk;off}} = 0.01$$
$$a^0_{\mathrm{Src}} = 0.0001, \quad a^0_{\mathrm{PTP}\alpha} = 0$$
$$p_1 = 0.05, \quad p_2 = 0.15, \quad p_3 = 0.035$$

Default initial conditions:

$$\mathrm{srci}_{t=0} = 1, \mathrm{cbp}_{t=0} = 1,$$
$$\mathrm{csk}_{t=0} = 1, \mathrm{PTP}\alpha_{t=0} = 1$$

all other species are 0 at $t = 0$

deactivation. Ubiquitination of active Src, causing protein degradation, has been suggested as an active such mechanism (Hakak and Martin, 1999).

The upper right part of Fig. 2 is concerned with PTPα. Src-directed activity of PTPα again depends on its phosphorylation status: Phosphorylation of Tyr789 seems to increase specific activity towards Src; and phosphorylation on serine residues decreases its inhibitory binding to Grb2 (Zheng *et al*., 2000). The hypothesised phosphorylation feedback loop between Src and PTPα is represented by interactions G, C and D in the model. We assumed that the two modifications, Tyr789 and serine phosphorylation, are independent of each other. The model therefore shows PTPα as two entities, '*PTP*α' and '*PTP*α*-pTyr*789'. Serine phosphorylation of PTPα and association with Grb2 are not explicitly represented in the form of molecular entities in the model. We can however simulate this effect through K_{ser}, which stands for the fraction of PTPα phosphorylated on serine residues and therefore immune to Grb2 inhibition.

The CSK/Cbp subsystem is depicted in the lower right-hand corner of Fig. 2. Src activity causes phosphorylation of the membrane protein Cbp. CSK can then associate with Cbp through a pTyr-SH2 interaction (node '*Cbp-P-CSK*'). This interaction both activates CSK and recruits it to the membrane, where it can interact with Src (Kawabuchi *et al*., 2000; Okada *et al*., 1991).

Total amounts of Src, CSK and PTPα are assumed to be constant. We are not aware of any evidence for differential expression of Src between cell cycle phases. PTPα has been shown to be expressed at constant levels in interphase and mitosis (Zheng and Shalloway, 2001). We have assumed constant overall CSK concentration, but we will later discuss the influence of CSK concentrations on Src regulation.

Some kinases and phosphatases in the system are unidentified, and their activities are denoted p_1 to p_3. p_1 stands for the activity of a phosphatase targetting Src pTyr418, while p_2 and p_3 target PTPα pTyr789 (phosphatase and kinase, respectively).

The interaction between CSK and Cbp was modelled as a simple association/dissociation reaction with $k_{\mathrm{Cbp;on}}$ and $k_{\mathrm{Cbp;off}}$ as forward and reverse reaction rates. The dissociation constant then becomes

$$K_d = \frac{k_{\mathrm{Cbp;off}}}{k_{\mathrm{Cbp;on}}}.$$

Bistability in the activation of Src

To study the effect of CSK on the system let us first consider a smaller subsystem of this model, where CSK concentration is kept constant (open-loop model). This can be realised by cutting interaction F or setting $k_{\mathrm{Cbp}} = 0$. CSK activity (represented by *Cbp-P-CSK*) now becomes a model parameter and can be varied independently.

Fig. 3 shows two phase-plane diagrams that demonstrate the response of this subsystem to variation in CSK activity. The lines indicate steady-state solutions (fixed points), where the system is at rest. Heavy lines represent stable solutions and dashed lines represent unstable solutions. When perturbed by a small amount, the system will return to a stable fixed point on either branch.

The two *saddle-node bifurcation* points SN1 and SN2 enclose a bistable region. For any CSK concentration within this region, the system has two stable solutions and one unstable solution. The system therefore exhibits *hysteresis*, which is characteristic of bistable systems: When travelling leftwards along the lower stable branch by decreasing CSK activity, Src remains inactive until SN2 is reached. If we decrease CSK any further, the lower stable fixed point disappears and the system moves towards the upper stable branch, corresponding to high Src activity. If CSK increases again, the system will proceed along the upper stable branch until SN1 is reached and another transition occurs, bringing the system back to the lower branch.

The two stable branches clearly separate high from low Src activity and this effect is even more pronounced for PTPα activity. These findings suggest that Src may be switched on and off through a bistable switch.

Role of the negative feedback loop involving CSK

We will now return to the full model and reactivate the negative feedback loop involving CSK translocation. The system is still attracted to the stable manifolds of Fig. 3. High Src activity, however, causes activation of CSK, which leads the system back to its original state.

This phenomenon is called *excitable behaviour* and is well-known in other disciplines of biological modelling. Examples are systems which involve spatiotemporal pattern formation or signal

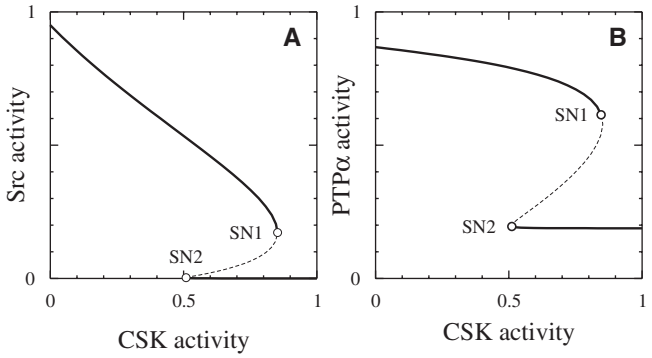


Fig. 3. The Src system exhibits bistability with respect to CSK. Lines show fixed point solutions, where the system is at equilibrium. Dashed lines indicate unstable solutions. Two saddle-node bifurcation points (SN1 and SN2) mark the borders of a bistable region.

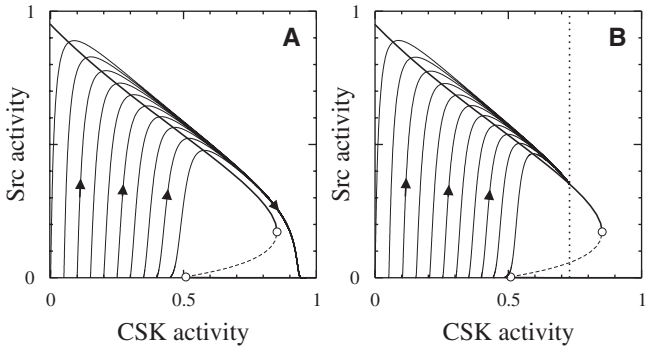


Fig. 4. Trajectories demonstrate the effect of negative feedback. **A**: Following a drop in CSK activity below SN2, the system is attracted by the upper stable branch. High Src activity now leads to activation of CSK, pushing the system to the right (i.e. towards high CSK) along the branch and restoring low Src activity. **B**: This process can be interrupted if the amount of CSK available to Cbp is limited (dotted line). In this case the limitation is imposed by simulating a weaker Cbp-CSK interaction ($K_d = 0.1$).

propagation, such as in the famous neural membrane potential model by Hodgkin and Huxley (1952).

The phase-plane diagram in Fig. 4A shows trajectories of the simulated system response to a drop in Src-directed CSK activity. For default parameters, excitation occurs below a threshold of CSK ≈ 0.5. After Src activation, CSK activity is restored through phosphorylation of Cbp by Src and translocation of CSK near the membrane. The trajectory leads back along the upper stable manifold to the original state on the lower branch.

This represents what we might observe during normal progression through the cell cycle: transient activation of Src at mitosis, followed by low Src activity after cytokinesis. However, our simulations show that this system is capable of more differentiated responses that could account for cell cycle checkpoint responses, in which Src activity is sustained at high level for longer periods.

We obtain a qualitatively different response if we restrict the amount of available CSK or increase the dissociation constant K_d for Cbp and CSK (Fig. 4B): As the system moves along the upper stable branch towards SN1, Src activity leads to complete conversion of *Cbp* into *Cbp-P*. Free Cbp is depleted before the system reaches the bifurcation point. It therefore comes to rest on the upper stable branch.

As we will demonstrate, several system parameters will allow us to achieve this kind of dynamic behaviour. This mechanism of obtaining sustained Src activity can be exploited by a regulatory system to produce diverse responses to environmental conditions.

Influence of physiological parameters

Let us examine the influence of the parameter k_2. It determines the rate at which Src is dephosphorylated on pTyr529 through the effect of PTPα, represented by v_1 and v_3 in Fig. 2 and Table 1. This parameter reflects the status of several Src phosphorylation sites located near the N-terminus. The cyclin-dependent kinase cdc2 (also known as cdk1) is associated with phosphorylation of these serine and threonine residues. When phosphorylated the cleft between the C-terminal tail and the SH2 domain widens and Tyr529 becomes more accessible to dephosphorylation. k_2 is therefore a function of cdc2 activity and thus an important connection to cell cycle.

Fig. 5A demonstrates the influence of k_2 on bistability of the system. As k_2 increases the bistable range expands and is shifted to higher CSK activities. Low k_2 values (corresponding to low cdc2 activity and therefore weak interaction between PTPα and Src) result in a strong inhibitory effect of CSK.

For a high k_2 the position of SN1 exceeds the amount of total available CSK in the cell (dotted line) and CSK activity is no longer sufficient to bring the system back to the lower stable branch. Any further increase in k_2 amplifies this effect. We can display this relationship between k_2 and Src activity at equilibrium in another bifurcation diagram (Fig. 5B), from which two new bifurcation points, SN3 and SN4 emerge.

To the left of SN3 we observe excitable behaviour due to strong negative feedback. Within the bistable region, CSK will initially keep Src activity low, but cannot restore the low Src state after excitation. Finally, SN4 represents the point, above which CSK by itself is not even capable of enforcing the low Src state and Src becomes constitutively active. A CSK deficient cell will generally display this phenotype (Imamoto and Soriano, 1993; Nada *et al*., 1993).

These results show that the system exhibits bistability on at least two different levels: activation of Src by CSK and control of excitability. Many other bistable signalling systems have been characterised to yield an all-or-nothing response similar to what can be seen in Fig. 3 (Bhalla and Iyengar, 1999; Laurent and Kellershohn, 1999). The system described here is capable of three qualitatively distinct types of behaviour: stable high, excitable and bistable.

We see a potential role for this new, bistable region in the G2/M cell cycle checkpoint. Src normally displays only transient activity during these cell cycle phases, but the regulatory system needs to guarantee that downstream events of Src are completed before proceeding to the next stage. For example, if delays occur due to unfavourable environmental conditions, the action of cdc2 will keep the system to the right of SN3. This means that Src activity will continue until cdc2 activity falls below this point. The low Src state, however, is robust to cdc2 variation: Src activation is only triggered

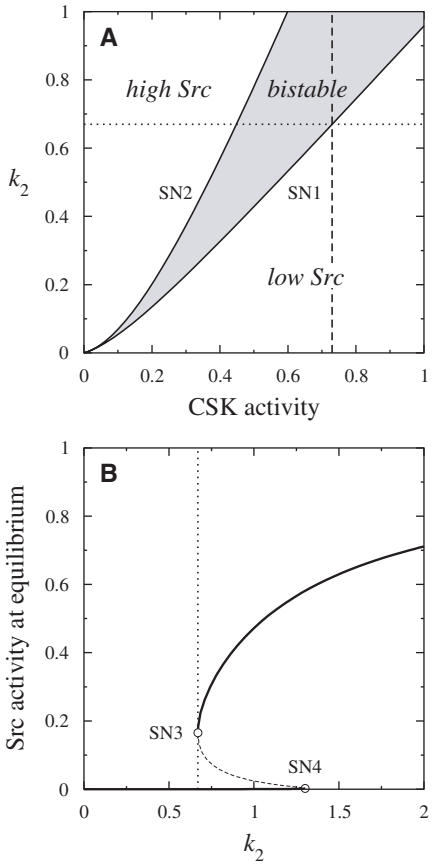


Fig. 5. Effect of k_2 on bistability. **A**: The bistable region expands with increasing k_2. The two straight lines containing the bistable region correspond to the positions of SN1 and SN2 from Fig. 3. The dashed line represents the amount of available CSK. At a critical value of $k_2 \approx 0.67$ SN1 crosses this line and the low-Src area is no longer reachable after Src activation (dotted line). **B**: In the full model, with the CSK negative feedback loop in place, we therefore observe another set of bifurcation points (SN3 and SN4). Sustained activation of Src is not possible to the left of SN3.

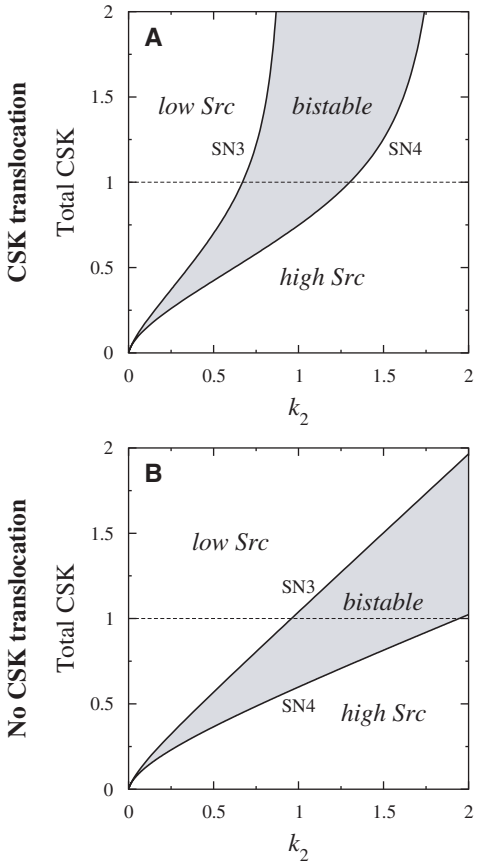


Fig. 6. The amount of available CSK controls the position of bifurcation points SN3 and SN4 in Fig. 5B. Default values are indicated as dashed lines. **A**: In the default model, higher amounts of CSK do not significantly disturb the system. **B**: In a hypothetical alternative model, in which CSK is activated by direct phosphorylation, the system becomes more sensitive to the abundance of CSK. This demonstrates the benefit of an indirect activation via translocation.

by a drop in CSK activity: CSK switches the system on, while cdc2 controls when it is switched off.

CSK translocation and robustness

As we have seen, the fact that the amount of available CSK is limited creates a second set of bifurcation points that define another bistable region. Experimental evidence suggests that the Src system is in fact sensitive to this amount (Imamoto and Soriano, 1993; Nada *et al*., 1993). Surprisingly, in our model we find instead a remarkable degree of robustness with respect to CSK.

Fig. 6 demonstrates the influence of total CSK on the bifurcation points SN3 and SN4 (see Fig. 5). Fig. 6A is based on the default parameter set. While lowering total CSK below the default of 1.0 results in a larger range of high Src activity, increasing total CSK has more modest effects. This is not surprising, as the amount of Cbp has been kept constant at 1 in this experiment. Greater concentrations of CSK result in saturation of phosphorylated Cbp, where more inactive CSK remains in the cytoplasm.

In order to establish an explanation for the mechanism in which CSK is assumed to be regulated, we have created a hypothetical, alternative version of our model, where Cbp is not involved and CSK is activated by direct Src phosphorylation. Fig. 6B shows the same bifurcation diagram for this alternative model. In this theoretical system the positions of the bifurcation points SN3 and SN4 are directly proportional to the total concentration of CSK. A two-fold increase in CSK would therefore shift the bistable area from $k_2 \in [0.67, 1.3]$ to [2.0, 4.4]. This is in contrast to the translocation model, where the bifurcation points asymptotically approach a finite value and the same increase has a weaker impact on the position of the bistable range ($k_2 \in [0.87, 1.74]$). The theoretical model thus displays greater sensitivity to CSK variation than the translocation model.

Translocation has the advantage that regulation depends on two protein concentrations instead of only one. Alterations to one of these have a smaller impact than in the simple activation model. However, experimental findings suggest that Src is in fact sensitive to singular variations of CSK. We will consider some explanations for this discrepancy under Discussion and Conclusion.

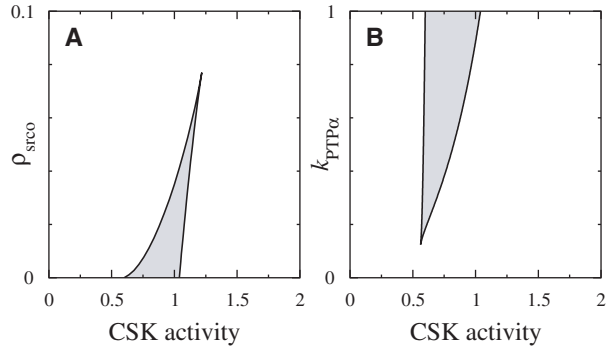


Fig. 7. Critical parameters for existence of bistability. Graphs show deviations from default parameter set, where $\rho_{srco} = 0$ and $k_{PTP\alpha=1}$. **A**: Participation of *srco* (ie not phosphorylated on Tyr418 nor Tyr529) in positive feedback removes bistability. **B**: External positive feedback via PTPα is required for bistability. Variation of k_2 and k_3 yields different limits, but the critical point remains significantly above zero within physiologically acceptable parameter ranges.

Prerequisites for bistability

Existence of bistability in the Src system depends on a number of model parameters. Surprisingly, our simulations show that an increased ρ_{srco} (i.e. activity of unphosphorylated Src) eliminates bistability (Fig. 7). For the default parameter set, the unstable branch disappears above $\rho_{srco} \approx 0.075$. However, a reasonable separation of branches is only achieved below $\rho_{srco} \approx 0.02$. This figure corresponds to at least a 50-fold increase of activity upon Tyr418 phosphorylation (ρ_{srca}/ρ_{srco}).

Due to autocatalytic activity it is difficult to experimentally correlate phosphorylation state of Tyr418 to Src activity. Under *in vitro* conditions autophosphorylation of wildtype Src cannot be eliminated completely. Kmiecik and Shalloway have used Src mutants possessing a phenylalanine substitution at Tyr418 (Y418F) to study the effects of autophosphorylation deficiency. Their *in vitro* kinase assays suggest a more modest, roughly five-fold increase in activity, corresponding to a ρ_{srco} of 0.2. However, they also show that the Y418F mutant has drastically lowered *in vivo* activity (Kmiecik and Shalloway, 1987, Table 1). The small bistable area of Fig. 7A suggests that activity of the unphosphorylated kinase domain is in fact nearly or completely absent.

The model also allows for an alternative explanation, which is that pTyr418 is required for some targets, but not for others. Strictly speaking any enzymatic rate is dependent on both enzyme and substrate. Kinase assays are usually carried out with small detectable peptide substrates. It is conceivable that larger protein substrates behave differently and that various conformations of Src show diminished (or possibly even enhanced) activity towards those substrates. The central '*Src activity*' node in our model admittedly is a simplification.

To further investigate this hypothesis we extended our model to account for substrate-specific Src activities. The extended model contains three different nodes that replace the '*Src activity*' node: Cbp-specific, PTPα-specific and autophosphorylation-specific activity. The simulation results show that only ρ_{srco} rates for PTPα-directed activity and autophosphorylation are critical for bistability. Above $\rho_{srco,\,auto} \approx 0.12$ the system becomes monostable, while variation of $\rho_{srco,\,Cbp}$ has no effect on stability (data not shown). In conclusion, our model suggests that the activity of Src in absence of Tyr418 phosphorylation is either entirely suppressed or at least ineffective on the two positive feedback loops.

Role of the positive feedback loops

Positive feedback is known to be an essential ingredient for bistability in biochemical systems (Cinquin and Demongeot, 2002; Angeli *et al*., 2004). The system described here provides a good example of sophisticated control over a positive feedback loop.

In fact, our model contains two positive feedback loops (see Fig. 1): Src autophosphorylation (B) and the PTPα loop (C). The strength of loop B is determined by the kinetic parameter k_3. Loop C depends on multiple parameters: k_2, K_{ser} and $k_{PTP\alpha}$. Similar to k_2, K_{ser} is a physiological parameter. K_{ser} is directly proportional to the activity of serine/threonine kinases that phosphorylate PTPα, inhibiting the interaction of PTPα and Grb2. Because of its position in the model, the behaviour produced by variation of K_{ser} is analogous to k_2.

As we have demonstrated in Fig. 5, the parameter k_2 is dependent on phosphorylation status of the N-terminal Src phosphorylation sites, which in turn are controlled by cdc2 activity. cdc2 effectively determines the amount of positive feedback that Src experiences after dephosphorylation on pTyr529. Thus, Fig. 5 gains a new interpretation: it demonstrates the influence of the external positive feedback loop (loop C) on Src activation and deactivation.

While decreasing k_2 confines bistability to a very small region, inhibiting Src-mediated tyrosine phosphorylation of PTPα has an even more dramatic effect on bistability (Fig. 7B). If $k_{PTP\alpha}$ falls below approx. 0.12, the bistable region disappears. Even an unreasonably large increase of k_3 – the parameter controlling feedback loop B – cannot reinstate bistability. The conclusion is that internal feedback is not sufficient to maintain bistability in the Src system.

DISCUSSION AND CONCLUSION

We have developed a dynamic systems model of a small set of interacting proteins centred around the protein tyrosine kinase Src. The model reproduces its activation during mitosis, as influenced by the tyrosine kinase CSK and tyrosine phosphatase PTPα. We have shown that the model is consistent with the observations made in normal, healthy cells, as well as in a number of perturbed systems such as CSK mutants.

The simulations reveal an interesting implementation of bistability at the base of an excitable system. Bistability is frequently associated with systems which display a sustained response to a short, transient signal, toggling between two discrete states. Excitable behaviour is less common in biochemical systems. The combination of these two elements produces a system capable of differentiated responses. Depending on the parameter k_2 we observe either full excitability, sustained Src activation or a bistable combination of both. Part of this complex behaviour may be required for biological phenomena such as cell cycle checkpoints.

If we accept bistability as an explanation for the complex manner in which Src is regulated, the model raises several questions. Bistability only exists if we assume that Src activity is completely or almost completely inactive under absence of phosphorylation of

Tyr418 in the activation loop. This must be true for at least PTPα-directed activity and for autophosphorylation. Kinase assays using small peptide substrates are probably not conclusive in this case.

Our simulations confirm that Src regulation is sensitive to the amounts of Cbp and CSK in the cell, although we observe a remarkable level of robustness to CSK overexpression, which is not observed *in vivo*. This discrepancy could indicate a role for other regulatory components in the system. There are several possible explanations: Src not only resides at membranes, but it also participates in cytoplasmic interactions. Even if regulation of membrane-bound Src was not affected by CSK overexpression, the detrimental effect of overexpression on cytoplasmic Src could be responsible for the observed lethality. Also it is not clear whether Cbp is increased in cell lines that overexpress CSK. If this is the case, this could indicate a regulatory influence of CSK on the expression of Cbp. The observed dynamics would be similar to Fig. 6B. Unfortunately, there is little data available about the Cbp/CSK subsystem, for example regarding expression levels of Cbp in wildtype and CSK overexpression cell lines or about its deactivation by tyrosine phosphatases. The resolution of this problem requires experimental evidence.

There is currently little supporting evidence for the existence of feedback loop C (Fig. 1) involving PTPα. Our results show that a feedback loop of this kind is likely to exist. Together with bistability critical characteristics of the system behaviour are lost without it. For example, control exerted by cyclin-dependent kinases (such as cdc2) is dependent on external positive feedback. These issues, too, will need to be addressed experimentally.

The behaviour we have reproduced probably demonstrates only a small part of what the tyrosine kinase Src is capable of. For example, it is known to also reside in the cytoplasm, or at other membranes than the cytoplasmic. Other phosphorylation sites influence its interactions and activity. PTPα is known to form inactive dimers, which could constitute a negative regulatory mechanism (Jiang *et al*., 2000). Src interacts with numerous other kinases, phosphatases, and transcription factors. Its regulation in another context is bound to be different from what we have described here. In order to understand the purpose of Src and kinase signalling in general the possibilities in these domains will need detailed exploration.

REFERENCES

Abram,C.L. and Courtneidge,S.A. Src family tyrosine kinases and growth factor signaling. *Experimental Cell Research*, 254(1):1–13, 2000.

Angeli,D., Ferrell,J.E. Jr., and Sontag,E.D. Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. *Proc Nat Acad Sci*, 101(7):1822–1827, 2004.

Bhalla,U.S. and Iyengar,R. Emergent properties of networks of biological signaling pathways. *Science*, 283(5400):381–387.

Bjorge,J., Jakymiw,A., and Fujita,D. Selected glimpses into the activation and function of Src kinase. *Oncogene*, 19(49):5620–35, 2000.

Boerner,R., *et al*. Correlation of the phosphorylation states of pp60/c-Src with tyrosine'kinase activity: the'intramolecular pY530-SH2 complex retains significant'activity if Y419 is phosphorylated. *Biochemistry*, 35(29):9519–25, 1996.

Brdička,T., *et al*. Phosphoprotein associated with glycosphingolipid-enriched microdomains (PAG), a novel ubiquitously expressed transmembrane adaptor "protein, binds the protein tyrosine kinase CSK'and is involved in regulation of T cell activation. *J Exp Med*, 191(9):1591–604, 2000.

Brown,M.T. and Cooper,J.A. Regulation, substrates and functions of src. *Biochim Biophys Acta*, 1287(2-3):121–49, 1996.

Cinquin,O. and Demongeot,J. Roles of positive and negative feedback in "biological systems. *C R Biol*, 325(11):1085–95, 2002.

den Hertog, J. Tracy,S., and Hunter,T. Phosphorylation of receptor protein-tyrosine phosphatase alpha on Tyr789, a binding site for the SH3-SH2-SH3 adaptor protein GRB-2 in vivo. *EMBO J*, 13(13):3020–32, 1994.

Doedel,E.J. Auto: a program for the automatic bifurcation analysis of autonomous systems. In *Proc 10th Manitoba Conf on Num Math and Comp*, volume 30, pages 265–284. Univ. of Manitoba, Winnipeg, Canada, 1981. URL http://indy.cs.concordia.ca/auto.

Ermentrout,G.B. XPPAUT 5.91. Website, 2005. URL http://www.math.pitt.edu/ bard/xpp/xpp.html.

Fuß,H. Dubitzky,W., Downes,C.S., and Kurth,M.J. Mathematical models of cell cycle regulation. *Brief Bioinform*, 6(2):163–77, 2005.

Hakak,Y. and Martin,G.S. Ubiquitin-dependent degradation of active Src. *Curr'Biol*, 9(18):1039–42, 1999.

Hodgkin,A.L. and Huxley,A.F. A quantitative description of membrane current'and its'application to conduction and excitation in nerve. *J Physiol*, 117(4):500–44, 1952.

Howell,B. and Cooper,J. Csk suppression of Src involves movement of Csk to sites of Src activity. *Mol Cell Biol*, 14(8):5402–11, 1994.

Imamoto,A. and Soriano,P. Disruption of the CSK gene, encoding a negative regulator of Src family tyrosine kinases, leads to neural tube defects and embryonic lethality in mice. *Cell*, 73(6):1117–24, 1993.

Ingolia,N.T. and Murray,A.W. The ups and downs of modeling the cell cycle. *Curr Biol*, 14:R771–R777.

Irby,R.B., *et al*. Activating Src mutation in a subset of advanced human colon cancers. *Nat Genet*, 21(2):187–90, 1999.

Jiang,G., den Hertog, J. and Hunter,T. Receptor-like protein tyrosine phosphatase alpha homodimerizes on the cell surface. *Mol Cell Biol*, 20(16):5917–29, 2000.

Kawabuchi,M., *et al*. Transmembrane phosphoprotein Cbp regulates the activities of Src-family tyrosine kinases. *Nature*, 404(6781):999–1003, 2000.

Kmiecik,T. and Shalloway,D. Activation and suppression of pp60/c-Src "transforming ability by mutation of its primary sites of tyrosine phosphorylation. *Cell*, 49(1):65–73, 1987.

Laurent,M. and Kellershohn,N. Multistability: a major means of differentiation and evolution in biological systems. *Trends Biochem Sci*, 24(11):418–422.

Mandel,J., Palfreyman,N., and Dubitzky,W. Modelling codependence in biological systems. *IEE Proc Systems Biol*, 153(5), 2006. In Press.

Milo,R., *et al*. Network motifs: Simple building blocks of complex networks. *Science*, 298:824–827, 25 Oct. 2002.

Mustelin,T. and Hunter,T. Meeting at mitosis: cell cycle-specific regulation of c-"Src by RPTPalpha. *Sci STKE*, 2002(115):PE3, 2002.

Nada,S., *et al*. Constitutive activation of Src family kinases in mouse embryos that lack CSK. *Cell*, 73(6):1125–35, 1993.

Okada,M., *et al*. CSK: a protein-tyrosine kinase involved in regulation of Src family kinases. *J Biol Chem*, 266(36):24249–52, 1991.

Rous,P. A sarcoma of the fowl transmissible by an agent separable from the tumor cells. *J Exp Med*, 13(4):397–411, 1911.

Shenoy,S., Chackalaparampil,I., Bagrodia,S., Lin,P., and Shalloway,D. Role of p34/cdc2-mediated phosphorylations in two-step activation of pp60/c-Src during mitosis. *Proc Nat Acad Sci*, 89(15):7237–7241, 1992.

Shenoy,S., *et al*. Purified maturation promoting factor phosphorylates pp60/c-Src at the sites phosphorylated during fibroblast mitosis. *Cell*, 57(5):763–74, 1989.

Stover,D., Liebetanz,J., and Lydon,N. Cdc2-mediated modulation of pp60/c-Src activity. *J Biol Chem*, 269(43):26885–26889, 1994.

Sun,G., Sharma,A., and Budde,R. Autophosphorylation of Src and Yes blocks their inactivation by Csk phosphorylation. *Oncogene*, 17(12):1587–95, 1998.

Thomas,S. and Brugge,J. Cellular functions regulated by Src family kinases. *Annu Rev Cell Dev Biol*, 13:513–609, 1997.

Tyson,J.J., Chen,K.C., and Novák,B. Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Curr Opin Cell Biol*, 15(2):221–231, 2003.

Xu,W., *et al*. Crystal structures of c-Src reveal features of its autoinhibitory mechanism. *Mol Cell*, 3(5):629–38, 1999.

Xu,W., Harrison,S.C., and Eck,M.J. Three-dimensional structure of the tyrosine kinase c-Src. *Nature*, 385(6617):595–602, 1997.

Zheng,X.M., Resnick,R.J., and Shalloway,D. A phosphotyrosine displacement mechanism for activation of Src by PTPalpha. *EMBO J*, 19(5):964–"78, 2000.

Zheng,X.M. and Shalloway,D. Two mechanisms activate PTPalpha during mitosis. *EMBO J*, 20(21):6037–49, 2001.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e166–e173
doi:10.1093/bioinformatics/btl249

Context-specific independence mixture modeling for positional weight matrices

Benjamin Georgi[1,*] and Alexander Schliep[1,*]

[1]Max Planck Institute for Molecular Genetics, Department of Computational Molecular Biology, Ihnestrasse 73, 14195 Berlin, Germany

ABSTRACT

Motivation: A positional weight matrix (PWM) is a statistical representation of the binding pattern of a transcription factor estimated from known binding site sequences. Previous studies showed that for factors which bind to divergent binding sites, mixtures of multiple PWMs increase performance. However, estimating a conventional mixture distribution for each position will in many cases cause overfitting.

Results: We propose a *context-specific independence* (CSI) mixture model and a learning algorithm based on a Bayesian approach. The CSI model adjusts complexity to fit the amount of variation observed on the sequence level in each position of a site. This not only yields a more parsimonious description of binding patterns, which improves parameter estimates, it also increases robustness as the model automatically adapts the number of components to fit the data.

Evaluation of the CSI model on simulated data showed favorable results compared to conventional mixtures. We demonstrate its adaptive properties in a classical model selection setup. The increased parsimony of the CSI model was shown for the transcription factor Leu3 where two binding-energy subgroups were distinguished equally well as with a conventional mixture but requiring 30% less parameters. Analysis of the human-mouse conservation of predicted binding sites of 64 JASPAR TFs showed that CSI was as good or better than a conventional mixture for 89% of the TFs and for 70% for a single PWM model.

Availability: http://algorithmics.molgen.mpg.de/mixture

Contact: georgi@molgen.mpg.de, schliep@molgen.mpg.de

1 INTRODUCTION

The reliable identification of putative transcription factor binding sites (TFBS) in genomic sequences is a problem of considerable importance for understanding gene regulation. The accepted approach is to formulate a mathematical representation of the binding pattern of a given factor based on collections of confirmed binding site sequences. This representation is subsequently used to score candidate sequences for occurrences of said pattern. The effectiveness of this approach depends on the models ability to accurately formalize the regularities found in the confirmed sites. Positional weight matrices (PWM) (Staden, 1984; Staden, 1989; Werner, 1999; Stormo, 1990; Stormo, 2000) are a statistical approach to modelling the factor-specific binding site composition. A PWM is derived from a multiple alignment of confirmed binding sites. For each position in the alignment a distribution over the four bases is estimated from the corresponding alignment column. Assuming independence between positions, this gives a probabilistic model of the binding site of a specific factor which subsequently can be used to score whether a DNA sequence contains a binding site for this factor (Hertz and Stormo, 1999; Levy and Hannenhalli, 2002).

However, this approach relies on two strong assumptions, namely that *all* positions within the site are independent and, more importantly, that all binding sites of a factor are slight variations of the *same* sequence. The former has been shown to be a simplification of biological reality for such examples as the Zinc finger motive (Wolfe *et al*., 1997) or the Mnt repressor (Man and Stormo, 2001). For the latter there is ample biological evidence to make it at least doubtful: It is well known that TFBS occur in clusters of functionally interacting transcription factors (TF) in promotor regions, so called transcriptional modules (Bolouri and Davidson, 2002; Ludwig *et al*., 1998; Thompson *et al*., 2004). A single factor may have many different interaction partners for different genes and it has been shown that the topology of these modules has an impact on the binding site sequences found for about nine thousand sites in S. cerevisiae (Bilu and Barkai, 2005). Also, it is known that a single change in a binding site can have profound effects on both the interaction behavior of a factor (Ptashne, 2004) or the level of induced gene expression (Williams *et al*., 2000). Moreover, in (Kotelnikova *et al*., 2005) the authors find increased levels of conservation for non-consensus binding site positions for 16 factors in 10 bacterial genomes, concluding that these sites are subject to evolutionary pressure. This gives further evidence for a level of biological complexity of binding site sequences beyond the ''single site'' hypothesis and motivates the development of more sophisticated methods.

This issue has received some attention in recent years. In (Barash *et al*., 2003) the authors successfully used subclasses of Bayesian networks for *de novo* motive discovery, among them mixtures of PWMs. More recently, in (Hannenhalli and Wang, 2005) binding sites have also been described as mixtures of PWMs. There it was shown, that a two component mixture model yielded improved conservation scores and higher expression coherence when compared to using a single PWM for a collection of 64 PWMs taken from the JASPAR data base (Sandelin *et al*., 2004).

However, the conventional mixture approach has severe drawbacks. First, it is an essentially unsolved problem to choose an appropriate number of mixture components, in particular if data is sparse and the classical model selection techniques (Akaike 1973; Schwarz, 1978) do not apply. In general too few components lead to suboptimal performance due to insufficient generalization, while, more

*To whom correspondence should be addressed.

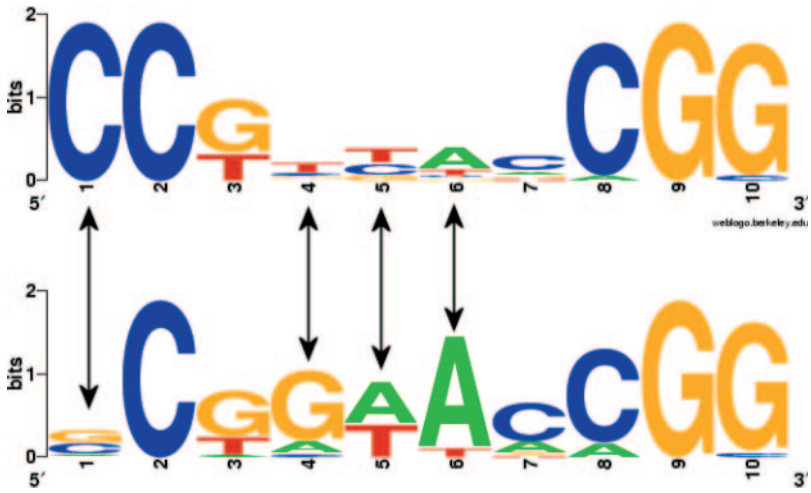


Fig. 1. WebLogos (*http://weblogo.berkeley.edu*) for the two subgroups of Leu3 binding sites. It can be seen that sequence variability is limited to positions 1, 4, 5 and 6 (indicated by arrows).

severely, too many components will cause overfitting. To circumvent this issue the number of components was fixed to two in (Hannenhalli and Wang, 2005). Secondly, it seems plausible that for most factors which have several types of binding sites (and can thus be modeled more precisely by a mixture), the different subgroups will not consists of distinct, dissimilar sequences. Rather, the variability between sites will be concentrated on specific positions. Estimating a full PWM for each mixture component will then introduce unnecessary parameters into the model. This increases model complexity unnecessarily and leads to less robust parameter estimates.

We present an extension of the conventional mixture framework that addresses these problems by learning an explicit dependency structure between the components of a PWM mixture. The basic idea of the method is to reduce the number of parameters required in the model by representing binding site positions with little variability in the different components by the same distribution. A biological example for such a situation is the TF Leu3. In (Hannenhalli and Wang, 2005) the authors showed that a two component mixture naturally separated the known binding sites (Liu and Clarke, 2002) into one high and one low binding-energy subgroup. Now, consider Fig. 1. The figure shows the sequence logos (Schneider and Stephens, 1990) for these subgroups. It can be seen that sequence variability is only present in position 1, 4, 5 and 6 (indicated by arrows) while the other sites are highly conserved. Another example is the factor Reb1. Reb1 binds with different affinities to motives TTACCCG and TTACCCT [37], that is the two subgroups differ in a single position only.

This notion of adapting model complexity to the data is known as *context-specific independence* (CSI) and has received considerable attention in the probabilistic reasoning community (Boutilier *et al*., 1996; Chickering and Heckerman, 1997; Friedman and Goldszmidt, 1998). In the context of mixture modeling, CSI has been successfully used for the analysis of gene expression data (Barash and Friedman, 2002).

The advantage of the CSI model in settings such as the Leu3 and Reb1 data is that in a conventional mixture random sequence deviations will cause the parameters in the different components for the same position to vary slightly, even if there is no meaningful variability on the sequence level. This overfitting introduces a distortion in the scores produced by the model that may result in a decrease in performance. Therefore, learning a CSI structure does not only yield a more parsimonious model, as less parameters are required, but also increases robustness for noisy data. Moreover, if components share the same group in the CSI structure for all positions, they can be merged thus reducing the number of components in the model. Therefore learning of a CSI structure allows for an automatic reduction of the number of components to a value more appropriate for a data set as an integral part of model training.

In the following sections we are going to introduce notation for the CSI mixture model and present the structure learning algorithm. We will then evaluate the performance of our method based on both simulated and real biological data.

2 METHODS

2.1 CSI mixture models

Before we begin defining the CSI mixture model we briefly introduce notation for conventional mixture models (refer to (McLachlan and Peel, 2000) for a detailed coverage of the subject). Let $X_1, \ldots, X_p$ denote random variables (RV) over the four bases (A,C,G,T) corresponding to a binding site with p positions. Given a data set D consisting of N samples x_i, $i = 1, \ldots, N$ where each x_i consists of an realization $x_{i1}, \ldots, x_{ip}$ of $X_1, \ldots, X_p$ a K component mixture distribution is given by

$$P(x_i) = \sum_{k=1}^{K} P(C = k) \prod_{j=1}^{p} P_j(x_{ij} \mid C = k), \tag{1}$$

where C is a RV representing the component number, the $P(C = k)$ are the component priors ($\sum_{k=1}^{K} P(C = k) = 1$) and the $P(x_{ij} \mid C = k)$ are discrete distributions over the four bases, conditional on the component RV C. That is, each $P(x_{ij} \mid C = k)$ is parameterized by a 4-component probability vector $\theta_{j|k}$. Define the collection of all $\theta_{j|k}$ and the weight vector $\theta_\pi = (P(C = 1), \ldots, P(C = K))$ as $\theta_M = (\theta_\pi, \theta_{j|k})$. Then θ_M completely parameterizes the mixture M. The likelihood $P(D \mid M)$ for data set D is simply the product over the mixture densities of each independent sample

$$P(D \mid M) = \prod_{i=1}^{N} P(x_i). \tag{2}$$

At this point we would like to point out that mixtures models and the extensions we are about to describe are not limited to discrete features. Rather the $P_j(x_{ij} \mid C = k)$ can be of any parametric family, be it

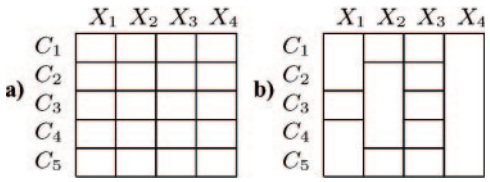


Fig. 2. a) Model structure for a conventional mixture with 5 components and four RV. Each cell of the matrix represents a distribution in the mixture and every RV has an unique distribution in each component. **b)** CSI model structure. Multiple components may share the same distribution for a RV as indicated by the matrix cells spanning multiple rows. In example C_2,C_3 and C_4 share the same distribution for X_2.

discrete or continuous and that in particular the domains of the X_j can be heterogenous.

In order to formally define the CSI mixture model it is helpful to first review the independence assumptions implicit in the conventional mixture model. In addition to the routine assumption of independence between the different data samples x_i, we also assume conditional independence between the X_j given a component k. This leads to a model structure as shown in Fig. 2a. The figure shows the structure matrix for a five component mixture with $p = 4$, each cell representing an uniquely parameterized distribution over the corresponding X_j. In a CSI model we qualify the general assumption of conditional independence between the X_j by representing multiple components with the same set of parameters. Essentially this amounts to learning a parameter tying structure for each X_j over the range of C. This is closely related to learning parameter ties in the topology of a *Hidden Markov Model* (HMM) (Durbin *et al*., 1998; Stolcke and Omohundro, 1994) as a mixture can be seen as an HMM with strongly constrained topology. In Fig. 2b we show one possible CSI structure for this model. Again, each cell of the matrix represents an uniquely parameterized distribution. This means that for example C_1 and C_2 are represented by the same distribution for X_1 and all components share the same distribution for X_4.

Formally we define the CSI mixture model as follows: For the set of component indexes $\mathcal{C} = \{1, \ldots, K\}$ and variables $X_1, \ldots, X_p$ let $G = \{g_j\}_{(j=1,\ldots,p)}$ be the CSI structure of the model M. Then $g_j = (g_{j1}, \ldots g_{jZ_j})$ where Z_j is the number of subgroups for X_j and each $g_{jr}, r = 1, \ldots, Z_j$ is a subset of component indexes from C. That is, each g_j is a partition of C into distinct subsets where each g_{jr} represents a subgroup of components which share the same distribution for X_j. The CSI mixture distribution is then obtained by replacing $P_j(x_{ij} \mid C = k)$ with $P_j(x_{ij} \mid g_j(k))$ in (1) where $g_j(k) = r$ such that $k \in g_{jr}$. Accordingly $\theta_M = (\theta_\pi, \theta_{X_j \mid g_{jr}})$ is the model parameterization. The complete CSI model M is then given by $M = (G, \theta_M)$.

The usefulness of the CSI approach for real world applications obviously depends on the ability to accurately and reliably determine an appropriate structure from data. This problem is addressed in the following section.

2.2 Structure learning

The task of learning a CSI model from data consists of assigning values to the group structure variables g_j and estimating parameters for the induced distributions. For the latter the *Expectation Maximization* (EM) (Dempster *et al*., and McLachlan and Krishnan, 1997) algorithm is the standard technique to arrive at parameter estimates. For the former, we adopted a Bayesian approach in the *Structural EM* algorithm framework (Friedman, 1997). This means that we score different candidate model structures based on the model posterior $P(M \mid D)$ which according to Bayes rule is given by

$$P(M \mid D) \propto P(M)P(D \mid M),$$

where $P(M)$ is a prior over the model structure and $P(D \mid M)$ is the Bayesian likelihood based on the data D and the *maximum a posteriori* (MAP) parameter estimates $\vec{\theta}_M$. That is

$$P(D \mid M) = P(D \mid \vec{\theta}_M)P(\vec{\theta}_M),$$

where $P(\vec{\theta}_M)$ is a prior over the model parameters in form of a product of conjugate Dirichlet priors over the individual elements of θ_M. The prior over the mixture weights θ_π was uniform, the priors over the $\theta_{X_j \mid g_{jr}}$ were chosen to be almost uniform with a small bias towards uniform θ (i.e., all hyper-parameters of the Dirichlets were set to 1.02). This was done to guard against overfitting by setting zero probabilities in the parameter estimation.

For the model prior $P(M)$ we adopted a fairly simple factored form

$$P(M) \propto P(K)P(G), \tag{3}$$

where the $P(K)$ is the prior over the number of components and $P(G)$ is the model structure prior. We set $P(K) = \gamma^K$ and $P(G) = \prod_{j=1}^{p} \alpha^{Z_j}$ with both γ and $\alpha < 1$. Thus by means of the prior we introduce a bias towards smaller models and simpler structures into the model posterior.

2.3 Learning algorithm

For a CSI mixture with K components over p RVs there are B_K^p possible model structures, where B_K is the Kth Bell number (Aigner, 1999). B_K gives the number of possible partitions of a set with K elements. This makes an exhaustive search over the structure space infeasible even for moderate sizes of K and p. For example for $K = 3$ and $p = 8$ there are 390,625 different structures. Instead we adopt an iterative greedy backward-selection procedure to learn a CSI model $M = (G, \theta_M)$. We initialize the procedure with $M^0 = (G^0, \theta_M^0)$, such that G^0 is the structure of maximal complexity (which is equivalent to a conventional mixture) and the initial parameters θ_M^0 are obtained by a single EM update based on a random assignment of data to components, followed by conventional parametric EM to obtain the MAP parameters.

In each following steps l we then use the current model $M^l = (G^l, \theta_M^l)$ to score the candidate structures G based on possible merges $(g_{jr}^l, g_{jz}^l) \rightarrow g_{jr}^l \cup g_{jz}^l$ $(r, z = 1, \ldots, Z_j, r \neq z)$ by computing the posteriors and accepting the candidate model with maximal posterior as M^{l+1}. Due to the independence assumption between the X_j we can score the candidate structures of each variable separately. In the framework of *Structural EM* (Friedman, 1997) this scoring can be done efficiently by computing the expected sufficient statistics of a candidate based on the current model M^l. Once we have determined G^{l+1} we can obtain the parameterization θ_M^{l+1} by running parametric EM. The procedure terminates when all candidate models have a posterior worse than M^l.

In summary, the structure learning procedure for an initial model M^0 consists of iterations over the following steps:

- Score possible candidates M^{l+1} based on M^l, accept candidate with maximal posterior.
- Optimize θ_M^{l+1} by running parametric EM.

2.4 Choosing the structure prior

One important aspect of the Bayesian approach to structure learning is the choice of the hyper parameters in the model prior. There are techniques for estimating these parameters directly from data (Robbins, 1956) or by simulation techniques such as Gibbs sampling (Gelman *et al*., 2003). In our application and for this first analysis we choose the structure prior parameter α directly based on a simple heuristic.

In general the prior $P(M)$ encodes the preference for a simpler model. This is contrasted with the data likelihood $P(D \mid M)$ which increases with model complexity. One way of thinking about the relation between prior and likelihood is that the prior acts as a regularization of the likelihood to prevent overfitting. From the perspective of the CSI structure learning task, the choice of the hyper parameter α of the structure prior $P(G)$ expresses our preference of a simpler, less complex structure. One way to look at this is that α puts a threshold on the decrease in likelihood we are willing to accept in exchange for a less complex structure. Since the likelihood of a data set is dependent on the sample size N the same must be true for α. To make this

explicit, consider the decision rule between a model M^l and a candidate model M during an iteration of the learning algorithm. Recall that M^l and M are identical except for a single merge in a g_j. This merge is accepted if

$$\frac{P(M^l \mid D)}{P(M \mid D)} = \frac{P(D \mid M^l)P(M^l)}{P(D \mid M)P(M)} \leq 1.$$

Substituting Eq. 2 and (3) and cancelling terms we obtain

$$\prod_{i=1}^{N} \frac{P(x_i \mid M^l)}{P(x_i \mid M)} \alpha \leq 1.$$

Each of the N fractions gives the decrease in likelihood of a x_i for moving from M^0 to the less complex model M. That is, we can think of each fraction as $(1 + \delta_i)$ where δ_i is the relative decrease in likelihood for x_i. Under the simplifying assumption that all of the δ_i are equal, i.e. $\delta_i = \delta$, we can now choose a δ as the *maximal relative decrease* in likelihood we are willing to accept in exchange for a less complex model. Then α is given by

$$\alpha = \alpha(\delta, N) = \frac{1}{(1+\delta)^N}.$$

It is important to stress that at this point all we have done is to replace the choice of α with the choice of δ. However this is advantageous for two reasons: First, the formula given above explicitly shows the impact of the data set size N. Secondly, δ has a straightforward interpretation as the reduction in likelihood between simple discrete distributions. As such it is easier to make an informed choice for δ based on the specific application. In our case it seemed reasonable to use a strong prior, such that the structure only introduced additional complexity into the model if clearly warranted by the data. In the following we chose the prior according to $\alpha(0.18, N)$ (unless noted otherwise). As an example for 20 sequences we obtain $\alpha(0.18, 20) = 0.036$.

2.5 Sequence scoring

One practical advantage of the model extensions described above is that it refines the models ability to represent TF binding patterns without abandoning the framework of probabilistic models. This means that the CSI model can be seamlessly and easily combined with established techniques for finding hits with significant scores in genomic sequences (Hertz and Stormo, 1999; Levy and Hannenhalli, 2002). Here, as in (Hannenhalli and Wang, 2005), the score of a mixture was defined as the maximum score over all components. This means that the score of a sequence was given by the strongest signal found among the components. Similar scoring schemes have been used for instance in the field of speech recognition.

3 RESULTS

3.1 Simulation studies

In order to examine the difference in performance between normal mixture and CSI models we generated artificial data sets from mixtures with differing numbers of components and structures.

In the first experiment the generating model was a two component CSI mixture with $p = 10$ and random weights θ_π. The CSI structure was set up as follows: Out of the ten positions, six were represented by single distributions in both components and four had a unique distribution in each component. The parameters of the distributions $\theta_{X_j \mid g_j}$ were chosen randomly.

First we evaluated the ability of our method to adapt to the structure in the data and thus to avoid overfitting. We trained one conventional and one CSI mixture model, both using three components on a training data set with 40 samples. The first result was that the structure learning algorithm recovered the generating models two component CSI structure with high accuracy (not shown). In order to quantify the advantage of the CSI model for sequence scoring we generated test data sets with 500 samples. We used a uniform background model to obtain the scores for each sample and the scores were then converted to p-values based on a score distribution on 1Mb of random sequence. We repeated the simulation for 30 different randomly generated data sets and observed that the CSI mixture yielded better (lower) p-values than the conventional mixture. The one-sided Wilcoxon test for paired samples assigned a significance of 0.02 to this result. Repeating the experiment with only 25 training samples confirmed these results with a Wilcoxon test significance of 0.04.

Table 1. Optimal model for the four data sets according to the average BIC over 30 repetitions

	Best model	Best avg. BIC
G_1	M_1	10851
G_2	M_2	11444
G_{CSI}	M_{CSI}	12266
G_4	M_{CSI}	12350

The next question we addressed was how the CSI model performed for different data sets in a classical model selection setup. We generated data sets of size 500 with $p = 10$ from four different models: a single PWM model G_1, a conventional two component mixture G_2, a CSI mixture with four components G_{CSI} and a conventional four component mixture G_4. The parameters of the discrete distributions in θ_M were chosen such that one base β was assigned a random probability sampled uniformly from [0.5,0.8] and the remaining mass split evenly over the other bases. In each case β was chosen such that it adhered to the CSI structure of the respective model, that is components that did not share a group for a X_j also had a dissimilar β. The structure in G_{CSI} consisted of 6 positions with four groups and two positions with three and two groups each. Subsequently we trained 30 models M of each of the four types (i.e. M_1, M_2, M_{CSI} and M_4) on each of the four data sets. Model fit was assessed by the *Bayesian Information Criterion* (BIC) (Schwarz, 1978). The best scoring model for each data set and its average BIC value based on the 30 repetitions is shown in Table 1. As one would expect, the model type that best matches the respective generating model yields the optimal BIC. A more interesting point to consider was the distributions of the differences of the remaining models to the optimal BIC shown in Fig. 3. It can be seen that for data sets where M_{CSI} is not optimal it achieves BIC scores very similar to the best. These results illustrate the inherent ability of CSI models to adapt to different data settings. This makes CSI a preferable choice of model for practical applications where the true number of components is unknown.

3.2 Analysis of TF LEU3

It was shown that 46 known binding sites of the TF Leu3 (Liu and Clarke, 2002) can be separated into a high and low binding-energy subgroup using a two component mixture with highly significant p-value (Hannenhalli and Wang, 2005). We repeated this analysis by training a two component CSI mixture. Since we were using the model in a clustering context a weak prior of $\alpha(0.05, 46) = 0.11$ was used. Fig. 4 shows the resulting CSI structure. Note the correspondence between the fully parameterized positions (1, 4, 5, 6) and the

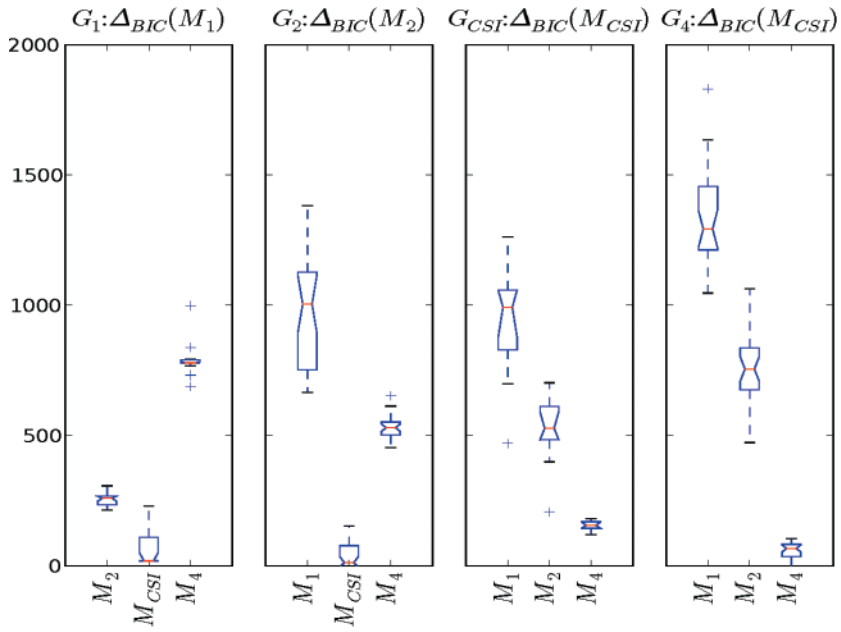


Fig. 3. Distributions of the difference in BIC to best scoring model for the four simulated data sets on 30 repetitions.

Fig. 4. Two component CSI mixture structure for known Leu3 binding sites. Each cell represents a discrete distribution, where cells spanning both rows identify positions with high conservation in both subgroups.

group specific sequence variability as visualized in Fig. 1. The CSI mixture yielded a subgroup division of the Leu3 sites that was practically identical to the one previously reported. However there are two important differences between the two models: First, the conventional mixture requires the estimation of 61 free parameters while due to the tying expressed in the CSI structure our model only needs 43 parameters. This means that CSI gave equivalent results using about 30% less parameters. Secondly, the CSI structure makes information about the subgroup and position specific sequence variability an explicit part of the model. Having this information readily available will facilitate further investigations, especially for large-scale studies where hundreds or more factors are involved.

3.3 Conservation statistics

The validation of predicted binding sites with respect to their biological functionality is a difficult problem as functionality cannot be directly assessed. One surrogate for functionality found in the literature is the degree of conservation in genomic sequences between related species (Thomas *et al*., 2003). For the sake of comparability with the results reported in (Hannenhalli and Wang, 2005) we follow the same evaluation approach taken there and evaluate the different models by the fraction of conserved predicted binding sites.

In the following we are going to evaluate the performance of a single PWM M_1, a two component mixture M_2 and a two component CSI mixture M_{CSI} based on human-mouse conservation. We used the same 64 JASPAR TFs as in (Hannenhalli and Wang, 2005). We downloaded the 1kb upstream regions of the **hg17** assembly (May 2004) from the UCSC genome data base (Hinrichs *et al*., 2006). The mouse conservation data (**mm7**) was extracted from the axtNet data set (Schwartz *et al*., 2003) (also UCSC). For each of the 64 TFs and each of the three models under consideration, we then computed the 1000 best scoring hits in the 1kb upstream regions. The overall base composition of the sequences was used as the background model. For the mixtures the hits were chosen proportionally to the mixing weights. This means that for a $\theta_\pi = (0.6, 0.4)$ we would chose the 600 best hits from the first component and the 400 best from the second. The fraction of hits that was conserved in mouse was then computed based on a 80% sequence identity cutoff.

Evaluation: In order to decrease the impact of random variation on the analysis we considered TFs with very similar fractions of conserved hits for two model types as not giving conclusive preference to any of the two. That is, if the difference in the conserved fraction was less than ten percent of the maximal conserved fraction observed for any of the three model types, the scores were considered to be ''equal'' for the purposes of this analysis. This has the effect of making the results more conservative in the sense that the impact of factors with very small differences in the conservation statistics was suppressed.

Fig. 5 shows the comparison of conserved fraction for the three model types. To illustrate the impact of the available number of training samples N for a factor on performance, we depict TFs differently based on the number of associated sequences. TFs with less than 18 sequences are shown as red diamonds, TFs with 19–31 sequences are shown as blue rectangles and TFs with more than 31 sequences are shown as green dots. The numbers were chosen as to split the 64 TFs into three roughly equally sized groups.

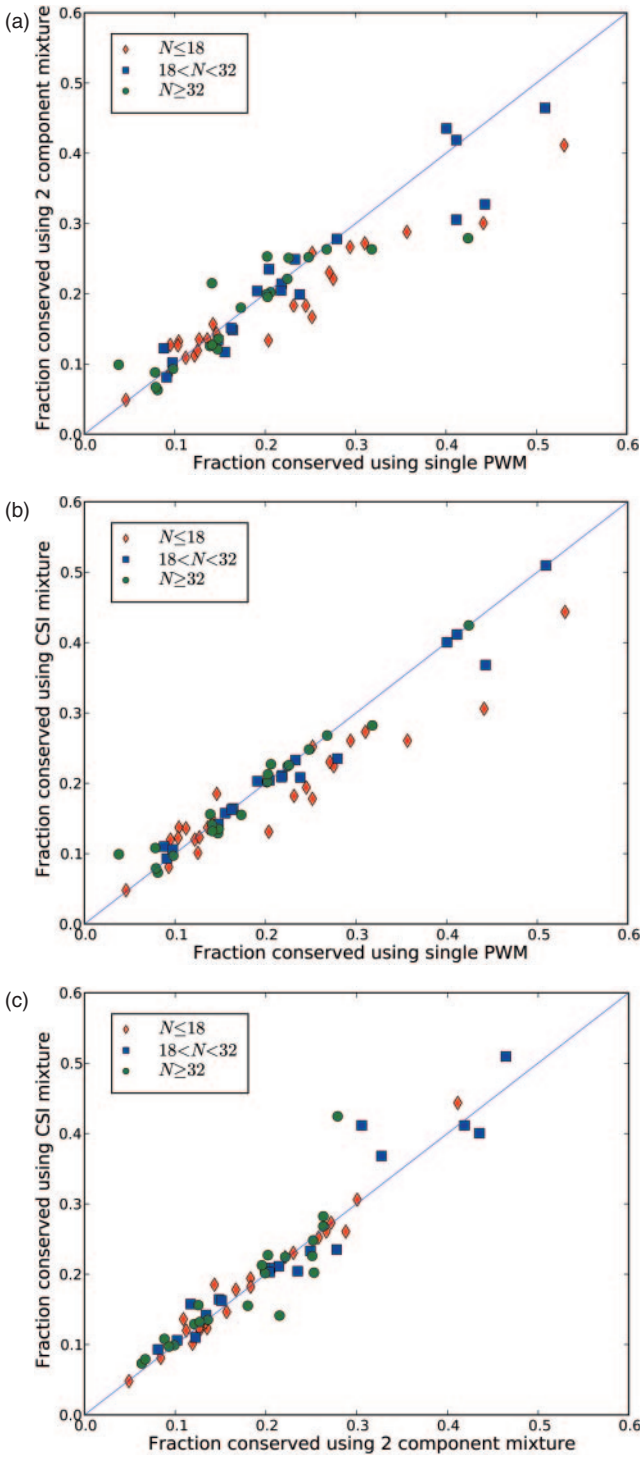


Fig. 5. a) Conserved fractions of hits for M_1 and M_2. The mixture M_2 is as good or better for 67% (43) of the TFs. **b**) Conserved fractions for M_{CSI} and M_1. For 70% (45) of the TFs the conservation of M_{CSI} was as good or better than for M_1. Outliers with strong preference for M_1 model had very few known sequences. If we only consider TFs with at least 20 sequences, the CSI yields as good or better conservation in 85% (34/40) of the cases. **c**) Comparison of conservation statistics of M_2 and M_{CSI}. For 89% (57) of the TFs M_{CSI} yields higher or equal conservation.

Table 2. Comparison of the conserved fraction of the 1000 best scoring hits for M_{CSI}, M_1 and M_2 in the two subsets of the TF data given the conditions ($M_2 \geq M_1$) and ($M_1 > M_2$) respectively

	$M_2 \geq M_1$ (43)	$M_1 > M_2$ (21)
$M_{CSI} \geq M_2$	84% (36)	100% (21)
$M_{CSI} > M_2$	47% (20)	81% (17)
$M_{CSI} \geq M_1$	89% (38)	33% (7)
$M_{CSI} > M_1$	37% (16)	10% (2)

M_1 **vs** M_2: In 5a) you can see the conserved fraction of M_1 and M_2 for the 64 TFs in the data set. The mixture model M_2 was as good or better than M_1 in 67% (43) of the cases. For 33% (21) of the TFs the mixture was strictly better. This means that the performance of the two component mixture was somewhat weaker in our analysis than reported in (Hannenhalli and Wang, 2005). Recall, that our data set differed from the one in (Hannenhalli and Wang, 2005) as it was based on a later genome freeze and, more importantly, it did not contain any downstream sequences. To the best of our knowledge the rest of our analysis was identical to the one conducted in (Hannenhalli and Wang, 2005).

M_{CSI} **vs** M_1: The comparison between the fraction of conserved hits of the CSI mixture M_{CSI} and the single PWM model M_1 can be seen in Fig. 5b). In 70% (45) of the TFs under consideration M_{CSI} showed a conserved fraction as good or better than M_1, with 28% (18) being strictly better. One important observation is that in most instances where M_1 had a strong advantage in conserved hits, the factor had only a small number of known binding sites. This can be seen by the large number of diamonds below the diagonal. For instance the rightmost point in Fig. 5b) at (0.53, 0.43) corresponds to MA0062 which has 7 known sites. In such a situation a little random variation in the sequences can have a strong impact on the trained model and lead to spurious structures. This is supported by the correlation between the number of available sequences for a factor and the increase in conservation for the CSI model. If we only considered TFs with 15 or more sequences, M_{CSI} is as good or better in 74% (40/54) of the cases, for 20 or more sequences in 85% (34/40) and for 40 or more in 94% (15/16). The fraction of TFs where M_{CSI} is strictly better remained in the range of 30% independent of the number of sequences.

M_{CSI} **vs** M_2: In Fig. 5c) we show the fraction of conserved hits for M_{CSI} and the conventional two component mixture M_2. For 89% (57) of the TFs the CSI model yields higher or equal conservation, 58% (37) being strictly greater.

Performance of M_{CSI}: Applying the two conditions ($M_2 \geq M_1$) and ($M_1 > M_2$) on the conserved fractions of hits split the 64 TFs in two subsets of size 43 and 21. We can think of the first subset as those TFs where a mixture model is appropriate and the second subset as being better represented by a single PWM. In the following we examined the performance of our CSI models within these two subsets. The results are summarized in Table 2. For the subset induced by ($M_2 \geq M_1$) M_{CSI} was as good or better then M_1 or M_2 for a strong majority of 84% (36) and 89% (38) of the TFs respectively. M_{CSI} was strictly better for 47% and 37% respectively. This means that for TFs where a two component mixture improves performance as compared to a single PWM, the CSI model will in most cases

outperform both of the other models. M_2 due to the reduction in overfitting and the more robust parameter estimates, M_1 because of the improved description of the binding pattern.

For the subset where a single PWM yielded a larger conserved fraction than the two component mixture (given by the condition $(M_1 > M_2)$) M_{CSI} was as good or better than M_2 for all the TFs in the subset (100% (19)) and strictly better for 81% (17). This illustrates the property of the CSI model to adapt to the number of subgroups supported by the data (one in this case) by means of the structure learning. M_{CSI} is equivalent or better than M_1 in 33% (7) of the TFs in the subset. This rather low number again shows the impact of spurious structures for TFs with few known binding sites. If we only consider the 11 TFs in the subset with 20 or more annotated binding sites, the value for $(M_{CSI} \geq M_1)$ goes up to 64% (7/11). Finally, M_{CSI} is strictly better than M_1 for a negligible 10% (2). This is not surprising as we would not expect CSI to outperform M_1 in situation where a single PWM is the appropriate model. Rather a successful application of the structure learning in such a case makes M_{CSI} equivalent to M_1. This corresponds to the points which lie directly on the diagonal (i.e. the conserved fractions are equal) in Fig. 5b).

4 DISCUSSION

The results of our simulation studies show that the CSI formalism yields more parsimonious and robust representations for TFs that exhibit a position-wise subgroup structure in their binding pattern. The greater parsimony of the CSI model as compared to conventional mixtures was demonstrated for a subgrouping of known Leu3 binding sites. In this example CSI required 30% less parameters than a conventional mixture for equal performance. The analysis of the conserved fraction of predicted binding sites in human upstream regions in mouse showed that a two component CSI model is clearly superior to a conventional two component mixture. This means that learning the CSI structures led to a more biologically meaningful characterization of the binding patterns of the TFs under consideration. For the TFs where the CSI model increased performance, we can assess that the known binding sites apparently exhibited a biologically relevant subgroup structure. The exact nature of the biological mechanisms underlying these subgroups remains elusive at this point. One possible explanation though would be the existence of different conformations of the TFs which show distinct binding patterns.

A strong advantage of the CSI (or conventional mixture) model over the single PWM model could not be observed on this data set. This was due to the occurrence of spurious structures for TFs with very few known binding sites and the large number of TFs where the single PWM model seems to be appropriate. This makes sense as one would expect the structure learning to be more vulnerable to outliers in situations where data is extremely sparse. The conclusion we draw from this result is twofold: First, CSI is a practical tool for the search for putative TFBS that fits in seamlessly within the probabilistic framework for scoring hits that has been established for the single PWM model (e.g. [18]). For a practical analysis using CSI though it seems important to require a minimum number of available binding sites (say 18) in order to attempt to fit a CSI model and to use the single PWM model otherwise. This could be easily included into the model prior. Secondly, we would expect the general usefulness of the CSI approach to increase in the future as the pool of known confirmed binding sites increases.

For future research we consider the development of more complex structure priors and improvements to the structure learning algorithm for sparse data. Also, it might be interesting to quantify the impact of different sequence scoring schemes on the performance the model. Moreover, since the probabilistic framework we work in is fully general, there are numerous biological applications where our method might yield improved results. In particular we consider applying our methods on donor splicing site detection, as larger data sets are available in this setting.

ACKNOWLEDGEMENTS

We would like to thank Martin Vingron and Gunnar Rätsch for helpful discussions and the reviewers for the excellent feedback they provided.

REFERENCES

[1] Aigner, M. A characterization of the Bell numbers. *Discrete Math*., 205:207–210, 1999.

[2] Akaike, H. Information theory and the extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, pages 267–281, 1973.

[3] Barash, Y., Elidan, G., Friedman, N., and Kaplan, T. Modeling dependencies in protein—dna binding sites. In *Proceedings of RECOMB '03*, pages 28–37, New York, NY, USA, 2003. ACM Press.

[4] Barash, Y. and Friedman, N. Context-specific bayesian clustering for gene expression data. *J Comput Biol*, 9(2):169–91, 2002.

[5] Bilu, Y. and Barkai, N. The design of transcription-factor binding sites is affected by combinatorial regulation. *Genome Biol*, 6(12):R103, 2005.

[6] Bolouri, H. and Davidson, E.H. Modeling DNA sequence-based cis-regulatory gene networks. *Dev Biol*, 246(1):2–13, 2002.

[7] Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. Context-specific independence in Bayesian networks. In *Uncertainty in Artificial Intelligence*, pages 115–123, 1996.

[8] Chickering, D.M. and Heckerman, D. Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. *Mach. Learn*., 29(2-3):181–212, 1997.

[9] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, pages 1–38, 1977.

[10] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. *Biological Sequence Analysis*. Cambridge University Press, 1998.

[11] Friedman, N. and Goldszmidt, M. Learning bayesian networks with local structure. In *Proceedings of the NATO Advanced Study Institute on Learning in graphical models*, pages 421–459, Norwell, MA, USA, 1998. Kluwer Academic Publishers, 1998.

[12] Friedman, N. Learning belief networks in the presence of missing values and hidden variables. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 125–133, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[13] Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. *Bayesian Data Analysis, 2nd edition*. CRC Press, 2003.

[14] Hannenhalli, S. and Wang, L.-S. Enhanced position weight matrices using mixture models. *Bioinformatics*, 21 Suppl 1:i204–i212, Jun 2005.

[15] Hertz, G.Z. and Stormo, G.D. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–577, July 1999.

[16] Hinrichs, A.S., Karolchik, D., Baertsch, R., Barber, G.P., Bejerano, G., Clawson, H., Diekhans, M., Furey, T.S., Harte, R.A., Hsu, F., Hillman-Jackson, J., Kuhn, R.M., Pedersen, J.S., Pohl, A., Raney, B.J., Rosenbloom, K.R., Siepel, A., Smith, K.E., Sugnet, C.W., Sultan-Qurraie, A., Thomas, D.J., Trumbower, H., Weber, R.J., Weirauch, M., Zweig, A.S., Haussler, D., and Kent, W.J. The UCSC Genome Browser Database: update 2006. *Nucleic Acids Res*, 34(Database issue):590–598, Jan 2006.

[17] Kotelnikova, E.A., Makeev, V.J., and Gelfand, M.S. Evolution of transcription factor DNA binding sites. *Gene*, 347(2):255–263, Mar 2005.

[18] Levy, S. and Hannenhalli, S. Identification of transcription factor binding sites in the human genome sequence. *Mamm Genome*, 13(9):510–514, Oct 2002.

[19] Liu, X. and Clarke, N.D. Rationalization of gene regulation by a eukaryotic transcription factor: calculation of regulatory region occupancy from predicted binding affinities. *J Mol Biol*, 323(1):1–8, 2002.

[20] Ludwig, M.Z., Patel, N.H., and Kreitman, M. Functional analysis of eve stripe 2 enhancer evolution in Drosophila: rules governing conservation and change. *Development*, 125(5):949–958, Mar 1998.

[21] Man, T.K. and Stormo, G.D. Non-independence of Mnt repressor-operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay. *Nucleic Acids Res*, 29(12):2471–2478, Jun 2001

[22] McLachlan, G.J. and Krishnan, T.J. *The EM Algorithm and Extensions*. Wiley, New York, 1997.

[23] McLachlan, G.J. and Peel, D. *Finite Mixture Models*. John Wiley & Sons, 2000.

[24] Ptashne, M. *A Genetic Switch: Gene Control and Phage l*. Cold Spring Harbor Laboratory Press, 2004.

[25] Robbins, H. An empirical bayes approach to statistics. In *Proc. Third Berkeley Symposium on Math. Statist. and Prob*., pages 157–164, 1956.

[26] Sandelin, A., Alkema, W., Engstrom, P., Wasserman, W.W., and Lenhard, B. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res*, 32(Database issue):91–94, Jan 2004.

[27] Schneider, T.D. and Stephens, R.M. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res*., 18:6097–6100, 1990.

[28] Schwartz, S., Kent, W.J., Smit, A., Zhang, Z., Baertsch, R., Hardison, R.C., Haussler, D., and Miller, W. Human-mouse alignments with BLASTZ. *Genome Res*, 13(1):103–107, Jan 2003.

[29] Schwarz, G. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[30] Staden, R. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res*, 12(1 Pt 2):505–519, 1984.

[31] Staden, R. Methods for calculating the probabilities of finding patterns in sequences. *Comput Appl Biosci*, 5(2):89–96, Apr 1989.

[32] Stolcke, A. and Omohundro, S.M. Best-first model merging for hidden Markov model induction. Technical Report TR-94-003, 1947 Center Street, Berkeley, CA, 1994.

[33] Stormo, G.D. Consensus patterns in DNA. *Methods Enzymol*, 183:211–221, 1990.

[34] Stormo, G.D. DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, Jan 2000. Historical Article.

[35] Thomas, J.W., Touchman, J.W., Blakesley, R.W., Bouffard, G.G, Beckstrom-Sternberg, S.M., Margulies, E.H., Blanchette, M., Siepel, A.C., Thomas, P.J., McDowell, J.C., Maskeri, B., Hansen, N.F., Schwartz, M.S., Weber, R.J., Kent, W.J., Karolchik, D., Bruen, T.C., Bevan, R., Cutler, D.J., Schwartz, S., Elnitski, L., Idol, J.R., Prasad, A.B., Lee-Lin, S.Q., Maduro, V.V., Summers, T.J., Portnoy, M.E., Dietrich, N.L, Akhter, N., Ayele, K., Benjamin, B., Cariaga, K., Brinkley, C.P., Brooks, S.Y., Granite, S., Guan, X., Gupta, J., Haghighi, P., Ho, S.L., Huang, M.C., Karlins, E., Laric, P.L., Legaspi, R., Lim, M.J., Maduro, Q.L., Masiello, C.A., Mastrian, S.D., McCloskey, J.C., Pearson, R., Stantripop, S., Tiongson, E.E., Tran, J.T., Tsurgeon, C., Vogt, J.L., Walker, M.A., Wetherby, K.D., Wiggins, L.S., Young, A.C., Zhang, L.H., Osoegawa, K., Zhu, B., Zhao, B., Shu, C.L., De Jong, P.J., Lawrence, C.E., Smit, A.F., Chakravarti, A., Haussler, D., Green, P., Miller, W., Green, E.D. Comparative analyses of multi-species sequences from targeted genomic regions. *Nature*, 424(6950):788–793, Aug 2003.

[36] Thompson, W., Palumbo, M.J., Wasserman, W.W., Liu, J.S., and Lawrence, C.E. Decoding human regulatory circuits. *Genome Res*, 14(10A):1967–1974, Oct 2004.

[37] Wang, K.L. and Warner, J.R. Positive and negative autoregulation of REB1 transcription in Saccharomyces cerevisiae. *Mol Cell Biol*, 18(7):4368–4376, Jul 1998.

[38] Werner, T. Models for prediction and recognition of eukaryotic promoters. *Mamm Genome*, 10(2):168–175, Feb 1999.

[39] Williams, J.R., Thayyullathil, C., and Freitag, N.E. Sequence variations within PrfA DNA binding sites and effects on Listeria monocytogenes virulence gene expression. *J Bacteriol*, 182(3):837–841, Feb 2000.

[40] Wolfe, S.A., Greisman, H.A., Ramm, E.I., and Pabo, C.O. Analysis of zinc fingers optimized via phage display: evaluating the utility of a recognition code. *J Mol Biol*, 285(5):1917–1934, Feb 1999.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e174–e183
doi:10.1093/bioinformatics/btl220

Improved Pruning algorithms and Divide-and-Conquer strategies for Dead-End Elimination, with application to protein design

Ivelin Georgiev[1], Ryan H. Lilien[1,2,3] and Bruce R. Donald[1,3,4,5,*]

[1]Dartmouth Computer Science Department, [2]Dartmouth Medical School, [3]Dartmouth Center for Structural Biology and Computational Chemistry, [4]Dartmouth Department of Chemistry and [5]Dartmouth Department of Biological Sciences, Hanover, NH 03755, USA

ABSTRACT

Motivation: Structure-based protein redesign can help engineer proteins with desired novel function. Improving computational efficiency while still maintaining the accuracy of the design predictions has been a major goal for protein design algorithms. The combinatorial nature of protein design results both from allowing residue mutations and from the incorporation of protein side-chain flexibility. Under the assumption that a single conformation can model protein folding and binding, the goal of many algorithms is the identification of the Global Minimum Energy Conformation (GMEC). A dominant theorem for the identification of the GMEC is Dead-End Elimination (DEE). DEE-based algorithms have proven capable of eliminating the majority of candidate conformations, while guaranteeing that only rotamers not belonging to the GMEC are pruned. However, when the protein design process incorporates rotameric energy minimization, DEE is no longer provably-accurate. Hence, with energy minimization, the *minimized-DEE* (*MinDEE*) criterion must be used instead.

Results: In this paper, we present provably-accurate improvements to both the DEE and MinDEE criteria. We show that our novel enhancements result in a speedup of up to a factor of more than 1000 when applied in redesign for three different proteins: Gramicidin Synthetase A, plastocyanin, and protein G.

Availability: Contact authors for source code.

Contact: Bruce.R.Donald@dartmouth.edu

1 INTRODUCTION

Desired novel protein function can result from the structure-based redesign of known protein sequences. In order to expedite the design process, a number of computational approaches for making redesign predictions have been successfully applied. In many protein design algorithms, the accuracy of the protein model is improved by incorporating protein flexibility (Street and Mayo, 1999; Jin *et al*., 2003; Jaramillo *et al*., 2001; Bolon and Mayo, 2001; Looger *et al*., 2003; Lilien *et al*., 2005). In (Najmanovich *et al*., 2000), a number of bound and unbound structures are compared, and the conclusion is drawn that only a small number of residues undergo conformational change, and that the structural changes are primarily side-chains, and not backbone. Hence, many protein design algorithms start with a rigid backbone conformation and optimize the residue sequence and the side-chain placements. Side-chain flexibility is typically modeled using a discrete set of low-energy rigid conformations, called rotamers (Lovell *et al*., 2000; Ponder and Richards, 1987). A major challenge for protein design algorithms is thus the combinatorial nature of the design process, resulting both from allowing residue mutations and from the incorporation of side-chain flexibility.

Under the assumption that a single conformation can accurately model protein folding and binding, the goal of many algorithms is the identification of the Global Minimum Energy Conformation (GMEC). It has been proven that protein design for a rigid backbone and using rotamers and a pairwise energy function is NP-hard (Pierce and Winfree, 2002; Chazelle *et al*., 2004). Hence, some heuristic approaches that do not make provable guarantees about the accuracy of the results have been developed (Street and Mayo, 1999; Kuhlman and Baker, 2000; Jin *et al*., 2003; Jaramillo *et al*., 2001; Marvin and Hellinga, 2001; Desmet *et al*., 2002; Shah *et al*., 2004). In contrast to such heuristic approaches (e.g., Monte Carlo, neural network, genetic algorithm), Dead-End Elimination (DEE) (Desmet *et al*., 1992; Lasters and Desmet, 1993) is a provable and efficient deterministic algorithm that is capable of eliminating the majority of the conformations, while guaranteeing that the GMEC is not pruned.

1.1 Traditional Dead-End Elimination

The DEE criterion (Desmet *et al*., 1992) uses rotameric energy interactions to identify and prune rotamers that are provably not part of the GMEC. The total energy of a conformation can be written as

$$E_T = E_{t'} + \sum_i E(i_r) + \sum_i \sum_{j>i} E(i_r, j_s). \quad (1)$$

Here, i_r specifies the particular rotamer identity r at residue position i; E_t' is the template energy (the energy of the rigid portion of the molecule); $E(i_r)$ is the self-energy (the intra-residue and residue-to-template energies) of rotamer i_r; and $E(i_r,j_s)$ is the non-bonded pairwise energy between rotamers i_r and j_s. In the original DEE criterion (Desmet *et al*., 1992), a *target* rotamer i_r could be provably

*To whom correspondence should be addressed.

pruned if a *competitor* rotamer i_t is found, such that the best (lowest) possible energy among conformations containing rotamer i_r is worse (higher) than the worst possible energy among conformations containing i_t. Hence, an alternative rotamer that is energetically more favorable than i_r exists for the entire conformation space, so i_r cannot be part of the GMEC and can thus be provably pruned. Formally, the DEE condition for pruning rotamer i_r is:

$$E(i_r) + \sum_{j \neq i} \min_s E(i_r, j_s) > E(i_t) + \sum_{j \neq i} \max_s E(i_t, j_s). \quad (2)$$

All the pairwise and self-energy terms are precomputed and a lookup is performed during the evaluation of the DEE condition. Eq. (2) is evaluated for each target rotamer i_r until either a superior competitor i_t is found and i_r can be pruned, or there are no unexamined competitors remaining, in which case i_r would not be pruned. For a protein with n residues and a maximum of q rotamers per residue, the complexity of evaluating Eq. (2) for all target rotamers is $O(q^2n^2)$.

The evaluation of Eq. (2) for all target rotamers represents a single *DEE pruning cycle*. Since rotamers that are pruned in a given cycle are not used in the evaluation of subsequent cycles, multiple repetitions of the DEE cycle can result in pruning a larger number of rotamers. Several extensions and enhancements to the original DEE criterion use more complex energy interactions and allow for additional pruning, at the cost of some additional complexity (Desmet *et al*., 1992; Lasters and Desmet, 1993; Goldstein, 1994; Gordon and Mayo, 1998; Pierce *et al*., 2000; Looger and Hellinga, 2001). Algorithms that combine several of these extensions into the DEE cycle significantly improve the pruning efficiency, thus allowing for the redesign of larger protein motifs (Gordon *et al*., 2003; Pierce *et al*., 2000). For a summary of DEE conditions, see Fig. 3(top).

The DEE pruning cycle can be repeated until the identification of the GMEC or until no more prunings are identified during a given cycle. Although DEE is a powerful algorithm, it does not guarantee a unique solution: multiple unpruned conformations may remain after pruning with DEE is exhausted. If DEE does not produce a unique conformation, the algorithm can report an unsuccessful design (Gordon *et al*., 2003; Pierce *et al*., 2000). As an alternative, the DEE pruning stage can be followed by an enumeration stage, in which the remaining conformations are examined and the GMEC is identified. In (Leach and Lemon, 1998), A^* branch-and-bound search is used after pruning with DEE to expand a conformation tree, so that conformations are extracted in order of conformational energy; the first conformation that is returned by the A^* search is the GMEC. The need to generate all unpruned conformations is thus eliminated, resulting in a combinatorial-factor reduction in the search space. However, since the enumeration stage is still exponential in nature, an efficient DEE pruning cycle is essential for making complex design problems computationally feasible.

1.2 Minimized Dead-End Elimination

Although rotamers represent low-energy side-chain conformations, the resulting discretization of the conformation space may decrease the accuracy of the underlying model (Desmet *et al*., 2002). The motivation for performing rotameric energy minimization is thus well-founded. However, when the protein design process incorporates energy minimization, DEE is no longer provably-accurate, since a pruned conformation may subsequently minimize to a lower energy than the energy of the DEE-identified GMEC. In (Georgiev *et al*., 2006), *MinDEE*, a novel generalized DEE algorithm is presented. In contrast to *traditional-DEE* (the DEE conditions described in Sec. 1.1), MinDEE guarantees that no rotamers belonging to the *minimized-GMEC* (*minGMEC*), the conformation with the lowest energy among all energy-minimized conformations, are pruned. Thus, in order to be provably-correct, MinDEE (instead of traditional-DEE) must be used for a design process that incorporates energy minimization.

In (Georgiev *et al*., 2006), it was experimentally confirmed that traditional-DEE can prune rotamers belonging to the minGMEC. For the 9-residue active site of the phenylalanine adenylation domain of the non-ribosomal peptide synthetase (NRPS) Gramicidin Synthetase A (GrsA-PheA) (PDB id: 1AMU) (Conti *et al*., 1997), traditional-DEE and MinDEE were applied in a 2-point-mutation redesign search[1] for switching the binding affinity of the protein from Phe to Leu. Traditional-DEE was shown to prune 2 of the 9 rotamers belonging to the minGMEC. Moreover, the energy of the minGMEC was approx. 5 kcal/mol lower than the energy of the *rigid-GMEC*.[2] The results in (Georgiev *et al*., 2006) thus confirm both that traditional-DEE is not provably-correct with energy minimization and that MinDEE is more capable of returning lower-energy (and hence, more stable) conformations.

The idea underlying MinDEE is analogous to the traditional-DEE approach: rotameric energy interactions are used to determine which rotamers are provably not part of the minGMEC and can be pruned. In contrast to traditional-DEE, however, since rotamers are allowed to energy-minimize, lower and upper bounds on the self- and pairwise rotamer energies must be used, instead of the rigid-energy terms $E(i_r)$ and $E(i_r, j_s)$ in Eq. (2). We will now describe the initial MinDEE criterion, closely following (Georgiev *et al*., 2006).

Without energy minimization, a rotamer stays in the same rigid conformation, independent of the rotamer identities for the remaining residues. In contrast, with energy minimization, a rotamer r at residue i may minimize from its initial conformation in order to accommodate a change from rotamer s to rotamer u at residue j. So that one rotamer does not minimize into another, rotameric movement is constrained to a voxel of conformation space. The voxel $\mathcal{V}(i_r)$ for rotamer i_r contains all conformations of residue i within $\pm\theta$ degrees around each rotamer dihedral. Similarly, the voxel for the pair of rotamers i_r and j_s is $\mathcal{V}(i_r, j_s) = \mathcal{V}(i_r) \times \mathcal{V}(j_s)$. The self-energy of a given rotamer can change as different conformations within the voxel are assumed. We can thus define the *maximum*, *minimum*, and *range* of voxel self-energies:

$$E_{\oplus}(i_r) = \max_{z \in \mathcal{V}(i_r)} E(z), \quad E_{\ominus}(i_r) = \min_{z \in \mathcal{V}(i_r)} E(z),$$
$$E_{\oslash}(i_r) = E_{\oplus}(i_r) - E_{\ominus}(i_r).$$

The *maximum*, *minimum*, and *range* of pairwise voxel energies are defined analogously (see Fig. 3). We now define the initial

[1] In a 2-point mutation search, any 2 of the 9 active site residues are allowed to mutate simultaneously.

[2] For clarity, we will henceforth call the GMEC returned by traditional-DEE, the *rigid-GMEC*.

MinDEE criterion as:

$$E_{\ominus}(i_r) + \sum_{j \neq i} \min_s E_{\ominus}(i_r, j_s) - \sum_{j \neq i} \max_s E_{\oslash}(j_s) - \sum_{j \neq i} \sum_{k \neq i,\, k > j} \max_{s,u} E_{\oslash}(j_s, k_u) > E_{\oplus}(i_t) + \sum_{j \neq i} \max_s E_{\oplus}(i_t, j_s). \quad (3)$$

If Eq. (3) holds, then there exists a competitor i_t whose worst possible conformational energy is lower than the best possible conformational energy for the target rotamer i_r. Hence, i_r cannot belong to the minGMEC and can be provably pruned (for a proof, see (Georgiev *et al*., 2006)). Eq. (3) for MinDEE is hence the analog of Eq. (2) for traditional-DEE. The most significant difference between traditional-DEE and MinDEE is the accounting for possible energy changes during minimization, which are incorporated through the introduction of the terms $\sum_j \max_s E_{\oslash}(j_s)$ and $\sum_j \sum_k \max_{s,u} E_{\oslash}(j_s, k_u)$. Similarly to traditional-DEE, the min and max self- and pairwise energy terms are precomputed and a lookup is performed during the pruning stage. Note that the terms $\sum_j \max_s E_{\oslash}(j_s)$ and $\sum_j \sum_k \max_{s,u} E_{\oslash}(j_s, k_u)$ can also be precomputed, since they are a function only of residue i. Thus, the MinDEE criterion (Eq. 3) can be computed as efficiently as the traditional-DEE criterion (Eq. 2).

The MinDEE criterion has been shown to be applicable both to GMEC-based and ensemble-based protein design (Georgiev *et al*., 2006). For the ensemble-based redesign, MinDEE was applied as a provable conformational-space filter in K^*, a scoring and search protein design algorithm that incorporates energy minimization (Lilien *et al*., 2005). Combined with A^* search, the Hybrid MinDEE-K^* algorithm introduced a significant improvement in computational efficiency over the original K^* results in (Lilien *et al*., 2005). In MinDEE/A^* (the GMEC-based algorithm), similarly to (Leach and Lemon, 1998) for traditional-DEE, MinDEE was first used to prune a large portion of the conformational space; the minGMEC was then extracted by A^* from the remaining conformations. Although MinDEE/A^* made the search for the minGMEC computationally feasible, the provable guarantees of the algorithm resulted in more conservative pruning and, hence, in slow running times (Georgiev *et al*., 2006). The derivation of novel techniques for improved pruning efficiency that can be incorporated into MinDEE/A^* is thus essential.

1.3 Contributions of the Paper

In this paper, we present novel provable enhancements both to traditional-DEE and MinDEE, for improved pruning efficiency. When applied in protein design searches, our enhancements yield a speedup of up to a factor of more than 1000. In particular, our paper makes the following contributions:

1. *DACS*: a provably-accurate divide-and-conquer enhancement to traditional-DEE. *DACS* is shown to obtain improved pruning efficiency and much faster running times. Due to its divide-and-conquer nature, *DACS* is especially beneficial in design problems where enumeration (Sec. 1.1) must be performed. The *DACS* algorithm is also extended to incorporate energy minimization.

2. *MinBounds*: a novel provable pruning criterion that incorporates energy minimization, generalizing the Bounds technique (Gordon *et al*., 2003) for protein design *without* energy minimization. MinBounds prunes all rotamers i_r for which the lower bound on the energy of all conformations that contain i_r is greater than a computed reference energy.

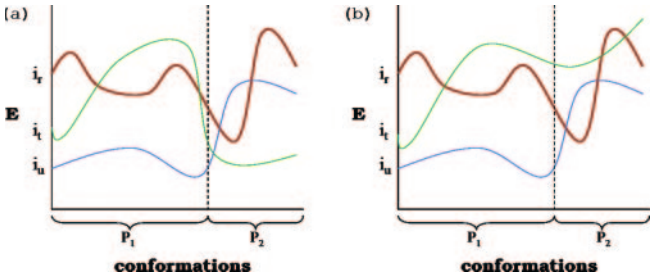


Fig. 1. Pruning with split-DEE and *DACS*. A point on the curve for rotamer i_r represents the energy of the corresponding conformation when residue i has the specific rotamer identity r. **(a)** Whereas the simple Goldstein criterion cannot prune i_r, conformational splitting can prune i_r by partitioning the conformational space. The dashed line shows a splitting of the conformational space into the two partitions P_1 and P_2. **(b)** Conformational splitting cannot prune rotamer i_r in partition P_2, so i_r must remain unpruned for the full conformational space. In contrast, *DACS* leaves i_r unpruned only for P_2; the local GMECs for P_1 and P_2 are computed and compared to obtain the overall GMEC. Note that the conformational space is discrete; continuity is shown here only for illustration purposes.

3. Analogously to the enhancements to traditional-DEE, we derive enhancements to the initial MinDEE criterion (Eq. 3) for additional pruning. The MinDEE analogs to the traditional-DEE simple and generalized Goldstein (Goldstein, 1994), conformational splitting (Pierce *et al*., 2000), and dead-ending pairs (Desmet *et al*., 1992; Lasters and Desmet, 1993) conditions are presented here; the simple Goldstein criterion was previously applied in (Georgiev *et al*., 2006).

4. A more efficient and powerful version of the MinDEE/A^* algorithm (Georgiev *et al*., 2006), incorporating MinBounds, *DACS*, and the enhancements to the initial MinDEE criterion. The new MinDEE/A^* algorithm is shown to lead to a significant improvement in pruning efficiency;

5. Application of our novel algorithms in GMEC-based searches for redesigning plastocyanin and the β1 domain of protein G, and for switching the substrate specificity of GrsA-PheA.

2 APPROACH

2.1 *DACS*

By partitioning the conformational search space, the original conformational splitting DEE (*split-DEE*) criterion (Pierce *et al*., 2000) (see Fig. 3g) enhances the pruning efficiency of traditional-DEE. Fig. 1a shows a simple example of the power of conformational splitting. In Fig. 1a, the simple Goldstein criterion ((Goldstein, 1994) and Fig. 3c) would not prune rotamer i_r, since it requires that there exist a competitor rotamer with better conformational energies than i_r for *all* conformations. In contrast, when split-DEE is used, the conformational space can be divided into several partitions, such that for each partition, there is some competitor that always has better conformational energies than i_r within that partition. In Fig. 1a, the dashed line divides the space into two partitions, P_1 and P_2. With this division, the competitor rotamer

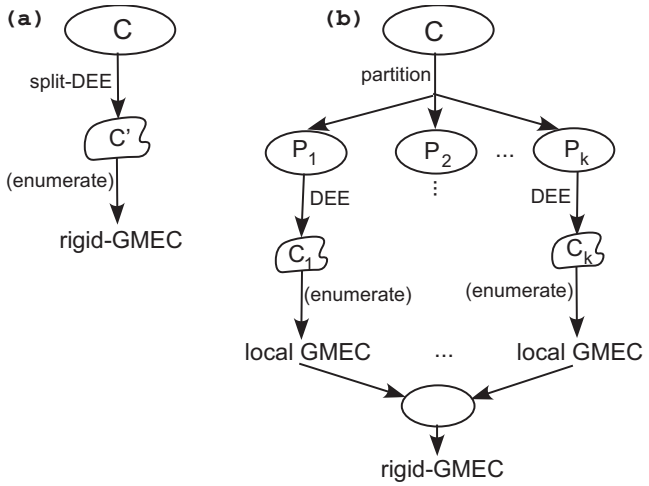


Fig. 2. Schematic of the (a) split-DEE, and (b) *DACS* algorithms. In **(a)**, the reduced set C' of conformations is obtained after split-DEE is applied to the initial conformational set C. If $|C'| = 1$, then split-DEE has output a unique solution, the rigid-GMEC; otherwise, enumeration must be performed. In **(b)**, the initial set C is first partitioned. DEE pruning is performed for each partition and the corresponding local rigid-GMEC is obtained. The lowest-energy conformation among the local GMECs for all partitions is the overall rigid-GMEC.

i_u always outperforms i_r in partition P_1, while rotamer i_t is always better than i_r in partition P_2. Thus, i_r can now be pruned, since there is always a better alternative for residue i, for any conformation. Hence, i_r is provably not part of the rigid-GMEC. The advantage of split-DEE is that no single competitor is required to outperform i_r for every conformation; as long as there exists a (different) dominant competitor for each partition, rotamer i_r can be pruned. A simplified schematic of split-DEE is given in Fig. 2a.

We now describe a modification of the split-DEE criterion that will allow for a further increase in pruning efficiency. Fig. 1b shows a different energy landscape. In this case, neither i_t nor i_u outperform i_r for all conformations in partition P_2. Thus, the original split-DEE criterion can no longer prune rotamer i_r and the potentially beneficial information that i_u is always better than i_r in partition P_1 is discarded. In general, it may be possible to prune i_r in the majority of the partitions, but so long as there exists a partition where no competitor is always better than i_r, the original split-DEE criterion must keep i_r unpruned. To remedy this loss of information, we relax the requirement that i_r be outperformed in all partitions; instead, we use a provably-accurate divide-and-conquer approach.

As in the original split-DEE criterion, we divide the conformational space into partitions. *Within* each partition, we apply DEE pruning to determine if there exists a competitor at residue i that always outperforms rotamer i_r. We then identify the *local* rigid-GMEC, restricted to the current partition, independently of the other partitions. If DEE pruning does not produce a unique solution, enumeration of the conformations in the current partition must be performed. The lowest-energy conformation among the local rigid-GMECs for all partitions is the *overall* rigid-GMEC (the rigid-GMEC among all conformations, for all partitions). We call this new approach *DACS* (**D**ivide-**A**nd-**C**onquer **S**plitting) (Fig. 2b). Note that in Fig. 1b, rotamer i_r is still unpruned in partition P_2, so the enumeration stage for P_2 must consider conformations containing i_r. However, in partition P_1, rotamer i_r can be provably pruned and hence all conformations in P_1 containing i_r can be eliminated from further consideration. With split-DEE, the conformations containing i_r for *both* partitions must still be enumerated. Hence, the general advantage of *DACS* over split-DEE is the ability to prune an additional combinatorial subset of the conformational space by exploiting partition-specific prunings.

The DEE pruning stage in *DACS* can incorporate any combination of the available provably-accurate traditional-DEE techniques (e.g., simple Goldstein and split-DEE). The enumeration stage is implemented using A^* search, which results in an additional combinatorial-factor reduction in the search space (see Sec. 1.1).

Several approaches based on ideas related to conformational splitting have been previously described. In (Looger and Hellinga, 2001), a generalized version of the split-DEE algorithm that is capable of pruning rotamer clusters, and not just single rotamers, was derived independently from (Pierce *et al*., 2000). A *split flags* technique was introduced in (Gordon *et al*., 2003) that is closely related to the approach in (Looger and Hellinga, 2001). With split flags, if a target rotamer i_r cannot be pruned for all partitions, the partitions in which i_r *can* be pruned are flagged as dead-ending. These split flags effectively represent dead-ending rotamer pairs.[3] Since the dead-ending pairs are not used in the evaluation of the DEE equations (e.g., Eq. 2), more single dead-ending rotamers may be identified in the subsequent DEE cycles.

Thus, both *DACS* and the split flags technique use pruning information that is otherwise discarded by split-DEE. However, there is one major advantage of the *DACS* algorithm over split flags, that can be attributed to the divide-and-conquer paradigm. Since the cost of expanding the A^* search tree depends combinatorially on the number of rotamers for each residue position,[4] a divide-and-conquer approach (in which the number of rotamers for each partition is reduced) can be more efficient than finding the global solution directly. Hence, for design problems in which the enumeration stage cannot be avoided, *DACS* should be especially useful.

In (Desmet *et al*., 1997), a divide-and-conquer algorithm for DEE pruning was described. In this algorithm, a list of dead-ending rotamers is constructed for each part of the divided conformational space; the *intersection* of all such lists gives the final list of pruned rotamers. Hence, this algorithm suffers from the same drawback as split-DEE: since a rotamer i_r cannot be pruned unless it is identified as dead-ending in all parts of the conformational space, potentially beneficial pruning information is often discarded.

The *DACS* algorithm benefits *both* from its divide-and-conquer nature and from the use of partition-specific prunings; *DACS* thus presents advantages over the other algorithms discussed in this section.

Correctness

We now prove the correctness of the *DACS* algorithm. Let C be the initial set of conformations and let q be the number of partitions P_i, $1 \leq i \leq q$, into which C is divided. Proposition 1 proves that *DACS* correctly identifies the local rigid-GMEC for each partition.

[3]In a dead-ending rotamer pair (i_r, j_s), either i_r or j_s may be part of the GMEC, but not both.

[4]For a protein with n residues and at most q rotamers per residue, the worst-case cost of expanding the A^* conformation tree is $O(q^n)$.

Proposition 2 shows that the overall GMEC is obtained as the lowest-energy conformation among the local GMECs, thus completing the proof of correctness for *DACS*.

PROPOSITION 1. *DACS identifies the local rigid-GMEC for each partition* P_i.

Proof: Let C_j denote the set of conformations for a given partition P_j, for an arbitrary j. Since the DEE pruning stage in *DACS* incorporates only provably-accurate traditional-DEE techniques the rigid-GMEC $g_j \in C_j$ is guaranteed not to be pruned. The rigid-GMEC for P_j is then extracted using the A^* search. □

PROPOSITION 2. *Let* g_i *be the local rigid-GMEC for partition* P_i *and let* $E(g_i)$ *be the total conformational energy of* g_i. *Then the overall rigid-GMEC is obtained as* $\text{argmin}_{gi}\ E\ (g_i)$, *for* $1 \leq i \leq q$:

Proof: We give a proof by contradiction. Let $h \in C$; $h \neq g_i$, $\forall i$, be the overall rigid-GMEC, so that $E(h) < \min_i E(g_i)$; that is, the overall rigid-GMEC h is not a local rigid-GMEC. By definition, h can be in exactly one partition of C; let this partition be P_j. It follows that $E(h) < E(g_j)$, so g_j is not the local rigid-GMEC for partition P_j. We thus have a contradiction. Hence, h must be a local GMEC; the lowest-energy local GMEC, $\text{argmin}_{g_i}\ E(g_i)$, for $1 \leq i \leq q$, is the overall rigid-GMEC. □

Partitioning

For each rotamer i_r, the original split-DEE (Pierce *et al*., 2000) forms partitions by choosing one or more of the protein residues as the *splitting positions* (*residues*).[5] Ideally, for n residues and s split positions, all $\binom{n-1}{s}$ possible combinations would be examined, until i_r can be pruned for all partitions in some combination. For $s > 2$, however, the increased algorithmic complexity suggests the use of a *magic bullet* approach to splitting (Gordon and Mayo, 1998). With this approach, a single combination (a magic bullet) of split positions is chosen, based on a heuristic ranking criterion.

In the original split-DEE, different rotamers can be pruned using different combinations of splitting residues, since the pruning information is combined *before* the enumeration stage of the search for the rigid-GMEC. *DACS* uses partition-specific pruning information, so the prunings for one partition are generally not valid for a different partition (see Fig. 1b). If different rotamers are pruned using different splitting residues, the divide-and-conquer-type approach can no longer be used. Thus, the *DACS* partitions must be identical for all rotamers tested for pruning. To partition the set of conformations, we therefore choose t split residues, $1 \leq t \leq n$, *before* applying the *DACS* criterion; we will henceforth refer to these split residues as *major* split residues, in contrast with the original split-DEE splitting positions.

We use a magic-bullet-type approach for choosing the *major* split residues. Assuming preliminary DEE pruning has been performed, we can rank residues in terms of the corresponding *p-ratio* (the ratio of pruned rotamers to total number of rotamers). The top t residues with the lowest p-ratio are chosen as the *major* split positions. Intuitively, residues with a low p-ratio are less prone to pruning and should thus minimize the cost of not being able to prune rotamers at the split positions.[6] Note that the method for choosing the *major* split residues does not affect the correctness of the algorithm, but may affect its pruning efficiency, so alternative methods for choosing the *major* split positions can also be applied.

[5]A *splitting position (residue)* divides the conformational space into partitions, such that each rotamer at that residue forms a separate partition.

Complexity

For t *major* split residues and at most q rotamers per residue, *DACS* divides the conformational space into $O(q^t)$ partitions. The cost of running the DEE cycle for each partition is determined by the complexity of the DEE algorithms in the cycle. As noted in Sec 1.1, the cost of the initial DEE criterion (Desmet *et al*., 1992) is $O(q^2n^2)$. The simple Goldstein criterion (Goldstein, 1994) has a complexity of $O(q^3n^2)$. An implementation of the original split-DEE (Pierce *et al*., 2000) with $s = 1$ split positions has the same complexity as simple Goldstein, assuming ($q > n$). The computation of split flags is done during the split-DEE run at no additional complexity. Hence, for a DEE cycle in which the most costly algorithm used is split-DEE, the general complexity of *DACS* is $O\left(q^{2+s+t}n\binom{n-1}{s}\right)$, where $O\left(q^{2+s}n\binom{n-1}{s}\right)$ is the cost of each split-DEE run. With $t = 1$ (a single magic bullet split position) for *major* splitting and $s = 1$ split-DEE in the inner loop, *DACS* runs in $O(q^4n^2)$, which is less than the cost of $s = 2$ split-DEE, $O(q^4n^3)$. Note that since the computation of the results for each partition is independent of the other partitions, *DACS* is easily parallelizable, which further reduces the effective complexity of the algorithm.

2.2 MinDEE Extensions

As already discussed, extensions to the initial traditional-DEE criterion have resulted in improved computational efficiency (Desmet *et al*., 1992; Lasters and Desmet, 1993; Goldstein, 1994; Pierce *et al*., 2000). Analogous MinDEE extensions for additional pruning are presented in Fig. 3. For example, the conformational splitting extension to MinDEE in Fig. 3(h) is the analog of the original split-DEE extension to traditional-DEE (Fig. 3g). The *DACS* algorithm is easily extended to incorporate energy minimization; in order to only prune rotamers that are provably not part of the *minGMEC*, the traditional-DEE criteria (Fig. 3, top) in the DEE cycle of *DACS* must be discarded and their MinDEE equivalents (Fig. 3, bottom) used instead.

2.3 MinBounds

We now present a provably-accurate pruning technique that is based on rotameric minimum energy bounds. The technique, *MinBounds*, is analogous to the Bounds approach of (Gordon *et al*., 2003) for traditional-DEE. In contrast to Bounds, however, MinBounds is provably-correct with energy minimization. Similarly to (Georgiev *et al*., 2006), we define the lower bound B_{i_r} on the *minimized* energy of all conformations containing rotamer i_r as:

$$B_{i_r} = E_{t'} + E_{\ominus}(i_r) + \sum_{j \neq i} \min_s E_{\ominus}(j_s) + \sum_{j \neq i} \min_s E_{\ominus}(i_r, j_s) + \sum_{j \neq i} \sum_{k \neq i,\, k > j} \min_{s,u} E_{\ominus}(j_s, k_u).$$

Thus, B_{i_r} is the best energy that a conformation can achieve after minimization if residue i has the particular rotamer identity r. Now,

[6]In each partition, there is only one rotamer for each *major* split residue.

Traditional-DEE

(a) $E(i_r) - E(i_t) + \sum_{j \neq i} \min_s E(i_r, j_s) - \sum_{j \neq i} \max_s E(i_t, j_s) > 0$ (Desmet et al., 1992)

(c) $E(i_r) - E(i_t) + \sum_{j,j \neq i} \min_s (E(i_r, j_s) - E(i_t, j_s)) > 0$ (Goldstein, 1994)

(e) $E(i_r) - \sum_{x=1,T} C_x E(i_{t_x}) + \sum_{j,j \neq i} \min_s \left(E(i_r, j_s) - \sum_{x=1,T} C_x E(i_{t_x}, j_s) \right) > 0$ (Goldstein, 1994)

(g) $E(i_r) - E(i_t) + \sum_{j, j \neq h \neq i} \left(\min_s (E(i_r, j_s) - E(i_t, j_s)) \right) + (E(i_r, h_v) - E(i_t, h_v)) > 0$ (Pierce et al., 2000)

(i) $E([i_r j_s]) - E([i_u j_v]) + \sum_{h \neq i,j} \min_t E([i_r j_s], h_t) - \sum_{h \neq i,j} \min_t E([i_u j_v], h_t) > 0$ (Desmet et al., 1992; Lasters and Desmet, 1993)

Minimized-DEE

(b) $E_\ominus(i_r) - E_\oplus(i_t) + \sum_{j \neq i} \min_s E_\ominus(i_r, j_s) - \sum_{j \neq i} \max_s E_\oplus(i_t, j_s) - \sum_{j \neq i} \max_s E_\oslash(j_s) - \sum_{j \neq i} \sum_{k \neq i, k > j} \max_{s,u} E_\oslash(j_s, k_u) > 0$ (Georgiev et al., 2006)

(d) $E_\ominus(i_r) - E_\oplus(i_t) - \sum_{j \neq i} \max_s E_\oslash(j_s) - \sum_{j \neq i} \sum_{k \neq i, k > j} \max_{s,u} E_\oslash(j_s, k_u) + \sum_{j \neq i} \min_s (E_\ominus(i_r, j_s) - E_\oplus(i_t, j_s)) > 0$

(f) $E_\ominus(i_r) - \sum_{x=1,T} C_x E_\oplus(i_{t_x}) - \sum_{j \neq i} \max_s E_\oslash(j_s) - \sum_{j \neq i} \sum_{k \neq i, k > i} \max_{s,u} E_\oslash(j_s, k_u) + \sum_{j \neq i} \min_s \left(E_\ominus(i_r, j_s) - \sum_{x=1,T} C_x E_\oplus(i_{t_x}, j_s) \right) > 0$

(h) $E_\ominus(i_r) - E_\oplus(i_t) - \sum_{j \neq i} \max_s E_\oslash(j_s) - \sum_{j \neq i} \sum_{k \neq i, k > j} \max_{s,u} E_\oslash(j_s, k_u) + \sum_{j \neq i, h} \left(\min_s (E_\ominus(i_r, j_s) - E_\oplus(i_t, j_s)) \right) + (E_\ominus(i_r, h_v) - E_\oplus(i_t, h_v)) > 0;$

(j) $E_\ominus([i_r j_s]) - E_\oplus([i_u j_v]) + \sum_{h \neq i,j} \min_t E_\ominus([i_r j_s], h_t) - \sum_{h \neq i,j} \max_t E_\oplus([i_u j_v], h_t) - \sum_{h \neq i,j} \max_t E_\oslash(h_t) - \sum_{h \neq i,j} \sum_{k \neq i,j,k > h} \max_{t,w} E_\oslash(h_t, k_w) > 0$

Fig. 3. Dead-End Elimination Pruning Conditions. A summary of the previously-described traditional-DEE pruning conditions (top) and our newly derived minimized-DEE pruning conditions (bottom). (a) is the initial criterion for traditional-DEE (Desmet *et al*., 1992), and (b) is the generalization for minimized-DEE (Eq. 3). The *simple* (d) and *general coupled* (f) minimized-DEE pruning conditions are analogous (resp.) to the corresponding *Goldstein* pruning conditions (c, e) of traditional-DEE (Goldstein, 1994). General Goldstein (e), in traditional-DEE, compares the energy of i_r to a weighted average of the interaction energies among T candidate pruning rotamers i_{t_x}. $C_x \geq 0$ is the weight given to the energy computed using rotamer i_{t_x}. The traditional conformational splitting criterion (Pierce *et al*., 2000) and the analogous MinDEE condition are given in (g) and (h), respectively. In the minimized-DEE generalization (j) of traditional Dead-Ending Pairs (i), $E_\diamond([i_r j_s]) = E_\diamond(i_r) + E_\diamond(j_s) + E_\diamond(i_r, j_s)(i \neq j), E_\diamond([i_r, j_s]), h_t) = E_\diamond(i_r, h_t) + E_\diamond(j_s, h_t)(i, j \neq h)$ where $E_\diamond \in \{E_\ominus, E_\oplus\}$.

let E_c be the minimized energy of a given conformation and E_g be the energy of the minGMEC, so that $E_c \geq E_g$. For a given rotamer i_r, if $B_{i_r} > E_c$, then $B_{i_r} > E_g$, so i_r cannot belong to the minGMEC and can thus be provably pruned.

In (Gordon *et al*., 2003), multiple Monte Carlo searches are used throughout the design process, in order to compute lower values for E_c (called the *reference energy*), so that more rotamers could be pruned by the Bounds criterion. Alternatively, in order to reduce the computational burden, MinBounds obtains E_c by energy-minimizing the wildtype only.

The MinBounds approach is most beneficial if used in a combination with the MinDEE criteria described in Sec. 2.2. Since the MinDEE conditions are conservative, a rotamer i_r cannot be pruned unless a better alternative is found, so some rotamers with bad (high) lower energy bounds may not be pruned by MinDEE. Using MinBounds with a good reference energy guarantees that rotamers with bad lower energy bounds will be pruned, further reducing the conformational search space.

Table 1. Traditional-DEE algorithms. The name of the algorithms (*left*) is shown with the corresponding sequence of pruning criteria (*right*). Each of the pruning criteria (as well as the full DEE cycle) is repeated until no further prunings are obtained. For each target rotamer i_r, *full* split-DEE attempts pruning for all possible combinations of $\binom{n-1}{s}$ split positions. The algorithms with *DACS* use $t = 1$ *major* split positions

SD_{1f}	Bounds, simple Goldstein, full $s = 1$ split-DEE;
SD_{2f}	Bounds, simple Goldstein, full $s = 1$ split-DEE, full $s = 2$ split-DEE;
SF_{2f}	Bounds, simple Goldstein, full $s = 1$ split-DEE w/ split flags, full $s = 2$ split-DEE w/ split flags;
DACS-SD_{1f}	SD_{1f}, followed by $t = 1$ *DACS* with a DEE stage incorporating the set of SD_{1f} criteria;
DACS-SD_{2f}	SD_{2f}, followed by $t = 1$ *DACS* with a DEE stage incorporating the set of SD_{2f} criteria;
DACS-SF_{2f}	SF_{2f}, followed by $t = 1$ *DACS* with a DEE stage incorporating the set of SF_{2f} criteria.

Table 2. Traditional-DEE redesign for GrsA-PheA (*a*), plastocyanin (*b*), and the β1 domain of protein G (surface) (*c*). The total number of conformations for cases (*a*), (*b*), and (*c*) is 4.78×10^{15}, 2.06×10^{27}, and 2.25×10^{22}, respectively. The *Enum* values show the number of remaining conformations after pruning with the algorithm given in the corresponding column; these conformations must be considered by A^* in the enumeration stage. *Time* shows the total running time (in minutes) consumed by each algorithm for the identification of the rigid-GMEC. All experiments were performed on a *single* processor.

		SD_{1f}	SD_{2f}	SF_{2f}	DACS-SD_{1f}	DACS-SD_{2f}	DACS-SF_{2f}
(*a*)	Enum	4.14×10^8	2.67×10^8	2.25×10^8	1.04×10^7	1.46×10^7	3.87×10^6
	Time	46.1	34.3	25.0	2.34	3.36	2.12
(*b*)	Enum	6.78×10^{12}	4.52×10^{12}	1.86×10^{12}	5.11×10^{11}	3.84×10^{11}	6.44×10^{10}
	Time	2057.1	1192.8	207.4	769.1	534.4	55.6
(*c*)	Enum	3.7×10^{12}	1.47×10^{11}	1.6×10^{10}	3.56×10^9	2.97×10^6	2.13×10^8
	Time	*	*	4540.2	171.3	6.5	154.1

* Did not complete in 10,000 minutes.

Table 3. Partition Pruning with *DACS-SF_{2f}* for GrsA-PheA. The conformational space was divided into 16 partitions by splitting at residue 322 (with a *p*-ratio of 29/45 after the initial pruning with SF_{2f}). The *Enum* values show the number of remaining conformations after pruning with DACS-SF_{2f}, for each of the 16 partitions. Due to rounding, these values do not sum exactly to the corresponding total number of conformations shown in Table 2.

	1	2	3	4	5	6	7	8
Enum	5.8×10^5	2.2×10^5	1.8×10^5	2.2×10^5	7.3×10^4	3.3×10^4	3.1×10^5	0.8×10^3
	9	10	11	12	13	14	15	16
Enum	1.1×10^3	2.0×10^4	5.3×10^3	1.1×10^4	2.8×10^5	4.5×10^5	4.4×10^4	1.5×10^6

3 ALGORITHMS

3.1 Traditional-DEE

The performance advantage of *DACS* for protein design without energy minimization is evaluated in comparison to the original split-DEE and split flags. The DEE pruning stage of the benchmarking algorithms is presented in Table 1. DACS-SD_{1f}, DACS-SD_{2f}, and DACS-SF_{2f} introduce an additional complexity factor of only $O(q)$, compared to, respectively, SD_{1f}, SD_{2f}, and SF_{2f} (see Sec. 2.1, **Complexity**). For all algorithms, the pruning stage is followed by an A^*-search enumeration stage.

3.2 MinDEE

We now present an improvement of the MinDEE/A^* algorithm (Georgiev *et al*., 2006), incorporating the simple Goldstein and conformational splitting extensions to the MinDEE criterion (Sec. 2.2), MinBounds (Sec. 2.3), and *DACS* for MinDEE (Sec. 2.2). In addition, the MinDEE/A^* algorithm is adapted to allow the use of the volume filter applied in the *ensemble-based* searches of (Lilien *et al*., 2005; Georgiev *et al*., 2006). The volume filter is applied to the initial set of mutation sequences.[7] pruning over- and under-packed sequences, relative to the original sequence. For each of the remaining sequences, the MinDEE analog of the DACS-SD_{1f} algorithm (Sec. 3.1) is used to eliminate the majority of the candidate conformations. A^* search is then applied in the enumeration stage to extract the minGMEC from the set of remaining conformations. Similarly to the *DACS* algorithm, the lowest-energy conformation among the rigid-GMECs for all mutation sequences is identified as the overall rigid-GMEC. If conformations within E_w of the minGMEC energy are to be generated, the pruning criteria and the A^* search can be modified accordingly (Georgiev *et al*., 2006). The application of the enhanced pruning conditions and the use of the volume filter aim at improving the pruning capabilities and the computational efficiency of the algorithm.

[7]A mutation sequence is a particular assignment of amino acid types for each residue.

4 METHODS

Structural Model. The NRPS enzyme GrsA-PheA (PDB id: 1AMU) (Conti *et al*., 1997) is used both for the traditional-DEE and MinDEE redesigns. Similarly to (Lilien *et al*., 2005; Georgiev *et al*., 2006), the residues modeled as flexible are the 9 active site residues (D235, A236, W239, T278, I299, A301, A322, I330, C331). In addition, our structural model consists of the steric shell (the 30 residues with at least one atom within 8 Å of a residue in the active site: 186Y, 188I, 190T, 210L, 213F, 214F, 230A, 234F, 237S, 238V, 240E, 243M, 279L, 300T, 302G, 303S, 320I, 321N, 323Y, 324G, 325P, 326T, 327E, 328T, 329T, 332A, 333T, 334T, 515N, and 517K), the amino acid substrate, and the AMP cofactor. The 9 flexible residues are allowed to mutate to the set (GAVLIFYWM) of hydrophobic amino acids. Traditional-DEE experiments are also performed on plastocyanin (PDB id: 2pcy) (Garrett *et al*., 1984). Based on (Gordon *et al*., 2003), we model as flexible 18 residues in the core of plastocyanin (5, 14, 21, 27, 29, 31, 37, 38, 39, 41, 72, 74, 80, 82, 84, 92, 96, 98), allowing them to mutate to the set (AVLIFYW) of hydrophobic amino acids. Similarly to (Gordon *et al*., 2003), redesign with traditional-DEE was also performed on 14 surface residues (4, 6, 8, 13, 15, 17, 42, 44, 46, 48, 49, 51, 53, 55) of the β1 domain of protein G (PDB id: 1pga) (Gallagher *et al*., 1994). The 14 residues modeled as flexible are allowed to mutate to the set (ANQSTDE); the remaining residues (except for the N-terminus) are modeled as part of the steric shell. Further, similarly to (Shah *et al*., 2004), 1pga redesign was performed on 12 core residues (3, 5, 7, 9, 20, 26, 30, 34, 39, 41, 52, 54), allowed to mutate to (GAVLIFYWM). **Rotamer Library.** Side-chain flexibility is modeled using the Richardsons' rotamer library (Lovell *et al*., 2000). **Energy Minimization.** Conformations are energy-minimized using steepest-descent

minimization and the AMBER energy function (electrostatic, vdW, and dihedral energy terms) (Weiner *et al*., 1984; Cornell *et al*., 1995). A voxel of $\theta = \pm 9^\circ$ is allowed around each rotamer dihedral. **Volume Filter.** *(MinDEE/A* only)* Over-/under-packed mutation sequences (by more than 30Å^3) relative to wildtype GrsA-PheA are pruned.

5 RESULTS AND DISCUSSION

Traditional-DEE. The results of applying the 6 different algorithms described in Sec. 3.1 to GrsA-PheA are shown in Table 2, Case (*a*). With $s = 1$ split-DEE (SD_{1f}), the redesign process took 46.1 minutes on a single processor, but the introduction of *DACS* (*DACS*-SD_{1f}) decreased the execution time by a factor of 20. For $s = 2$ split-DEE without and with split flags (SD_{2f} and SF_{2f}, respectively), the application of *DACS* resulted in a speedup factor of approx. 10 and 12, respectively. Thus, the minor additional complexity of the algorithms incorporating *DACS* (see Sec. 3.1) is outweighed by a significant increase in computational efficiency over the corresponding algorithms without *DACS*. Moreover, *DACS* performed better even when compared to more costly algorithms: *DACS*-SD_{1f} was a factor of 10 faster than the SF_{2f} algorithm.

A major factor for the speedup associated with the *DACS* algorithms is the corresponding increase in pruning efficiency (Table 2). By using a divide-and-conquer approach to partition the conformational space and identify partition-specific prunings, *DACS* allows for additional elimination, after pruning with the original split-DEE and split flags techniques is exhausted. Table 3 shows the *DACS*-SF_{2f} pruning results for all 16 partitions. As can be seen from Table 3, the remaining conformations after the DEE stage of *DACS* differ widely for each partition, ranging from less than 1,000 (partition 8) to approx. 1.5 million (partition 16). This variation shows that a different subset of rotamers can be pruned for each of the partitions, confirming the significance of using the *DACS* partition-specific prunings.

The improved execution times of the *DACS* redesigns can further be explained by the reduced cost of expanding the A^* search trees for each partition, resulting from the divide-and-conquer approach, as opposed to expanding the single A^* tree for the full conformational space. For example, for the SF_{2f} algorithm, A^* must simultaneously consider all of the remaining 2.25×10^8 conformations, whereas the largest partition for *DACS*-SF_{2f} has only 1.5×10^6 candidate conformations.

Table 2, Case (*b*), shows the plastocyanin redesign results for the six different algorithms used. Similarly to GrsA-PheA, the *DACS* algorithms (columns $4 - 6$) outperform the corresponding split-DEE/split flags algorithms in columns $1 - 3$, resulting in a speedup of up to a factor of 4. Unlike GrsA-PheA, however, the execution time for SF_{2f} was less than that for *DACS*-SD_{1f}, although the total number of unpruned conformations for *DACS*-SD_{1f} was smaller. We can thus conclude that the overhead of expanding separate A^* trees for each partition can be outweighed only by a significant improvement in pruning efficiency. However, in all of the redesign results presented in Table 2, the addition of the *DACS* algorithm (columns $4 - 6$) shows the necessary substantial increase in pruning efficiency over the *respective* algorithms (without *DACS*) in columns $1 - 3$. Hence, we conclude that, in general, *DACS* should be used as an enhancement, and not a substitute, to the other available DEE techniques.

Table 4. MinDEE/A^* Redesign for GrsA-PheA using $\mathbf{MA}_{new}$ (*a*) and $\mathbf{MA}_{simple}$ (*b*). The number of conformations remaining after the volume filter is 1.7×10^8. *Pruned* shows the number and percentage (in parentheses) of conformations pruned by the MinDEE stage of the corresponding algorithm; the number of remaining unpruned conformations is shown in *Remaining*; *Minimized* represents the number of conformations generated by A^* and energy-minimized. *Time*/*Seq*. is the average CPU time (in minutes) for the evaluation of a single mutation sequence.

	(*a*)	(*b*)
Pruned	1.697×10^8 (99.8%)	1.66×10^8 (97.6%)
Remaining	3.86×10^5	4.0×10^6
Minimized	9.3×10^4	9.64×10^4
Time/Seq.	16.11	16.66

The core redesign of the $\beta 1$ domain of protein G was completed within 5 minutes by all six algorithms (data not shown), which precludes a differential performance comparison for this case. However, our conclusions so far are confirmed by the (more difficult) surface redesigns of $\beta 1$ of protein G (Table 2, Case *c*). When compared to the algorithms without *DACS*, the respective *DACS* algorithms show a speedup of up to three orders of magnitude. In fact, the SD_{1f} and SD_{2f} algorithms exceeded the maximum allotted time of 10,000 minutes, so the use of *DACS* for these redesigns was essential. Moreover, similarly to Case (*a*), *DACS*-SD_{1f} performed an order of magnitude better than the more costly SF_{2f}.

Note that SF_{2f} in Case (*c*) ran 20 times slower than SF_{2f} in Case (*b*), although the number of unpruned conformations for Case (*c*) was two orders of magnitude lower. This is a direct result of the expansion mechanism of A^* and implies that, in order to generate the best conformation, a larger portion of the A^* conformation tree had to be expanded for SF_{2f} in Case (*c*) than in Case (*b*). Indeed, the A^* tree in Case (*c*) contained approx. 1.9×10^6 nodes at the time of completion, whereas the Case (*b*) tree contained only 5×10^5 nodes.

Also note the increased running time of *DACS*-SD_{2f} as compared to *DACS*-SD_{1f} (Case *a*) and *DACS*-SF_{2f} as compared to *DACS*-SD_{2f} (Case *c*). This can be explained by the choice of an inefficient *major* splitting residue. To test this hypothesis, we examined a different heuristic for choosing the *major* splitting positions, so that preference is given to lower-numbered residues.[8] With the new approach, a higher-numbered residue n_{i+k} is chosen as the *major* split position if its *p*-ratio is at least a value of α lower than the *p*-ratio of the lower-numbered residue n_i, $1 \leq i \leq n$. In our experiments, we used $\alpha = 0.15$. The new splitting approach significantly reduced the running times for most *DACS* redesigns (data not shown). *DACS*-SD_{2f} and *DACS*-SD_{1f} in Case (*a*) ran in 2.15 and 2.04 minutes, respectively. The running time of *DACS*-SD_{2f} (Case *c*) remained unchanged, whereas that of *DACS*-SF_{2f} was reduced by a factor of 44 to a total of 3.5 minutes. We can thus conclude that more sophisticated alternatives for choosing the *major* splitting positions should further improve the computational efficiency.

[8]Lower-numbered residues are at lower depths of the A^* conformation tree and are thus expanded first.

The results in this section show the additional pruning power and computational speedup of the *DACS* algorithm for traditional-DEE design, compared to the original split-DEE and split flags techniques, thus confirming the significance of this new approach.

MinDEE/A^*. Results from a 2-point mutation redesign search with energy minimization for switching the binding affinity of GrsA-PheA from Phe to Leu are shown in Table 4. Our improved version of the MinDEE/A^* algorithm[9] (Sec. 3.2), Table 4(a), is compared against the original MinDEE/A^* algorithm[10] (Georgiev *et al*., 2006), Table 4(b), which uses only the MinDEE analog of the simple Goldstein criterion. In order to fairly evaluate the effects of using the novel pruning criteria presented in this paper, the original MinDEE/A^* algorithm was also modified to incorporate the volume filter described in Sec. 3.2. For our experiments, we used a value of 6.0 for E_w (Sec. 3.2). The redesigns were performed on a cluster of 36 processors.

Only 30% of the mutation sequences passed the volume filter. The application of the MinDEE criteria in MA_{new} resulted in the elimination of (99.8%) of the remaining conformations, while the same algorithmic stage in MA_{simple} eliminated only (97.6%). The number of remaining conformations that had to be considered by A^* in the enumeration stage was consequently an order of magnitude smaller for the MA_{new} algorithm. Thus, as desired, the incorporation of the novel pruning techniques significantly enhanced the pruning capabilities of the MinDEE stage.

When considering the execution times, however, the speedup resulting from the use of the MA_{new} algorithm was not significant. The reason that the increased pruning efficiency did not lead to increased computational efficiency can be explained by the role of the MinDEE stage in the MinDEE/A^* algorithm. By pruning the majority of the possible rotamers, MinDEE reduces the cost of expanding the A^* search tree.[11] Since the number of rotamers for a *single* mutation sequence is comparatively small, the overhead of expanding the A^* tree is also smaller. Hence, for a single sequence, the execution time will be dominated mostly by the conformational energy minimization, and not by the tree expansion. Since an approximately equal number of conformations are energy-minimized by both MA_{new} and MA_{simple}, the similar execution times of both algorithms are not surprising. However, the fact that the novel advanced pruning techniques resulted in a significant increase in pruning efficiency, leads to the conclusion that the improved MinDEE/A^* algorithm will be especially useful in redesigns of larger systems[12] with energy minimization where the cost of managing the search tree dominates the computational effort.

6 CONCLUSION

In this paper, we presented novel enhancements for increased pruning efficiency, applicable in protein design problems both with and without energy minimization. The additional pruning power and the divide-and-conquer nature of the *DACS* algorithm were shown to lead to a significant computational speedup over other conformational-splitting-based algorithms, for the redesigns of GrsA-PheA, plastocyanin, and $\beta 1$ of protein G. Plastocyanin and protein G redesigns were also described in (Pierce *et al*., 2000; Gordon *et al*., 2003), using conformational splitting techniques in a combination with other advanced pruning criteria, such as dead-ending pairs. It would thus be interesting to incorporate such advanced pruning techniques into the *DACS* algorithms, in order to facilitate the faster design of larger systems. Moreover, since the choice of *major* splitting residues was shown to impact the efficiency of the algorithm, a further improvement of *DACS* could involve the derivation of a better approach for choosing the split positions. For larger systems, the use of multiple *major* split positions should also prove beneficial.

Our improved MinDEE/A^* algorithm incorporated the MinBounds technique, the simple Goldstein and split-DEE extensions to MinDEE, and the MinDEE version of *DACS*, resulting in a significant improvement in pruning efficiency over the original MinDEE/A^* algorithm. Similarly to traditional-DEE, further improvements to MinDEE/A^* could include the incorporation of $s = 2$ split-DEE and the split-flags techniques, as well as other advanced pruning criteria. As suggested by our results, in order to benefit from the increased pruning efficiency, MinDEE/A^* should be applied to larger systems, where the cost of expanding the search tree in the enumeration stage, rather than the energy minimization, will dominate the computation. MinDEE experiments on larger systems are currently under way and will be reported in future work.

The pruning techniques presented in this paper add to the power of available protein design algorithms and can be an important step towards the development of algorithms for the efficient solution of increasingly more computationally-expensive design problems. More efficient algorithms will also allow the use of improved models (e.g., larger rotamer libraries, improved energy functions, and the incorporation of backbone flexibility), thus increasing the accuracy of the design predictions.

[9]For convenience, we will henceforth refer to the this version as MA_{new}.

[10]Henceforth referred to as MA_{simple}.

[11]As noted before (Sec. 2.1), this cost depends combinatorially on the number of rotamers for each residue position.

[12]For example, larger proteins, a larger number of flexible residues, or the simultaneous redesign of multiple mutation sequences.

ACKNOWLEDGEMENTS

We thank Prof. A. Anderson, Dr. S. Apaydin, Mr. J. MacMaster, Mr. A. Yan, Mr. B. Stevens, and all members of the Donald Lab for helpful discussions and comments. This work is supported by grants to B.R.D. from the National Institutes of Health (R01 GM-65982), and the National Science Foundation (EIA-0305444).

REFERENCES

Bolon,D. and Mayo,S. (2001) Enzyme-like proteins by computational design. *PNAS USA*, **98**, 14274–14279.

Chazelle,B., Kingsford,C. and Singh,M. (2004) A semidefinite programming approach to side-chain positioning with new rounding strategies. *INFORMS Journal on Computing, Computational Biology Special Issue*, **16**(4), 380–392.

Conti,E., Stachelhaus,T., Marahiel,M. and Brick,P. (1997) Structural basis for the activation of phenylalanine in the non-ribosomal biosynthesis of Gramicidin S. *EMBO J*., **16**: 4174.

Cornell,W., Cieplak,P., Bayly,C., Gould,I., Merz,K., Ferguson,D., Spellmeyer,D., Fox,T., Caldwell,J. and Kollman,P. (1995) A second generation force field for the simulation of proteins, nucleic acids and organic molecules. *JACS*, **117**, 5179–5197.

Desmet,J., Maeyer,M., Hazes,B. and Lasters,I. (1992) The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, **356**, 539–542.

Desmet,J., Maeyer,M.D. and Lasters,I. (1997) Theoretical and algorithmical optimization of the dead-end elimination theorem. In Altman,R., Dunker,A., Hunter,L. and Klein,T. (eds), *Pac Symp Biocomput*., 122–33.

Desmet,J., Spriet,J. and Lasters,I. (2002) Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins*, **48**, 31–43.

Gallagher,T., Alexander,P., Bryan,P. and Gilliland,G.L. (1994) Two crystal structures of the B1 immunoglobulin-binding domain of streptococcal protein G and comparison with NMR. *Biochemistry*, **33**, 4721–4729.

Garrett,T.P., Clingeleffer,D.J., Guss,J.M., Rogers,S.J. and Freeman,H.C. (1984) The crystal structure of poplar apoplastocyanin at 1.8-a resolution. The geometry of the copper-binding site is created by the polypeptide. *J. Biol. Chem.*, **259**, 2822–2825.

Georgiev,I., Lilien,R. and Donald,B.R. (2006) A novel minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm for computing partition functions over molecular ensembles. In *Proceedings of The Tenth Annual International Conference on Research in Computational Molecular Biology (RECOMB),* pp. 530–545. Venice, Italy. Springer Berlin, RECOMB 2006, Lecture Notes in Computer Science, LNBI 3909.

Goldstein,R. (1994) Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophys. J.*, **66**, 1335.

Gordon,D., Hom,G., Mayo,S. and Pierce,N. (2003) Exact rotamer optimization for protein design. *J Comput Chem*, **24**, 232–243.

Gordon,D. and Mayo,S. (1998) Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J. Comput. Chem.*, **19**, 1505–1514.

Jaramillo,A., Wernisch,L., Héry,S. and Wodak,S. (2001) Automatic procedures for protein design. *Comb. Chem. High Throughput Screen.*, **4**, 643–659.

Jin,W., Kambara,O., Sasakawa,H., Tamura,A. and Takada,S. (2003) De novo design of foldable proteins with smooth folding funnel: Automated negative design and experimental verification. *Structure*, **11**, 581–591.

Kuhlman,B. and Baker,D. (2000) Native protein sequences are close to optimal for their structures. *PNAS*, **97**, 10383–10388.

Lasters,I. and Desmet,J. (1993) The fuzzy-end elimination theorem: correctly implementing the side chain placement algorithm based on the dead-end elimination theorem. *Protein Eng.*, **6**, 717–722.

Leach,A. and Lemon,A. (1998) Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins*, **33**, 227–239.

Lilien,R., Stevens,B., Anderson,A. and Donald,B.R. (2005) A novel ensemble-based scoring and search algorithm for protein redesign, and its application to modify the substrate specificity of the Gramicidin Synthetase A phenylalanine adenylation enzyme. *Journal of Computational Biology*, **12**(6–7), 740–761.

Looger,L., Dwyer,M., Smith,J. and Hellinga,H. (2003) Computational design of receptor and sensor proteins with novel functions. *Nature*, **423**, 185–190.

Looger,L. and Hellinga,H. (2001) Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: Implications for protein design and structural genomics. *J. Mol. Biol.*, **307**, 429–445.

Lovell,S., Word,J., Richardson,J. and Richardson,D. (2000) The penultimate rotamer library. *Proteins*, **40**, 389–408.

Marvin,J. and Hellinga,H. (2001) Conversion of a maltose receptor into a zinc biosensor by computational design. *PNAS*, **98**, 4955–4960.

Najmanovich,R., Kuttner,J., Sobolev,V. and Edelman,M. (2000) Side-chain flexibility in proteins upon ligand binding. *Proteins*, **39**(3), 261–8.

Pierce,N., Spriet,J., Desmet,J. and Mayo,S. (2000) Conformational splitting: a more powerful criterion for dead-end elimination. *J. Comput. Chem.*, **21**, 999–1009.

Pierce,N. and Winfree,E. (2002) Protein design is *NP*-hard. *Protein Eng.*, **15**, 779–782.

Ponder,J. and Richards,F. (1987) Tertiary templates for proteins: Use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.*, **193**, 775–791.

Shah,P., Hom,G. and Mayo,S. (2004) Preprocessing of rotamers for protein design calculations. *J Comput Chem*, **25**, 1797–1800.

Street,A. and Mayo,S. (1999) Computational protein design. *Structure*, **7**, R105–R109.

Weiner,S., Kollman,P., Case,D., Singh,U., Ghio,C., Alagona,G., Profeta,S. and Weiner,P. (1984) A new force field for molecular mechanical simulation of nucleic acids and proteins. *J. Am. Chem. Soc.*, **106**, 765–784.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e184–e190
doi:10.1093/bioinformatics/btl230

Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks

Olivier Gevaert[1,*], Frank De Smet[1,2], Dirk Timmerman[3], Yves Moreau[1] and Bart De Moor[1]

[1]Department of Electrical Engineering ESAT-SCD, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium, [2]Medical Direction, National Alliance of Christian Mutualities, Haachtsesteenweg 579, 1031 Brussel, Belgium and [3]Department of Obstetrics and Gynecology, University Hospital Gasthuisberg, Katholieke Universiteit Leuven, Herestraat 49, 3000 Leuven, Belgium

ABSTRACT

Motivation: Clinical data, such as patient history, laboratory analysis, ultrasound parameters—which are the basis of day-to-day clinical decision support—are often underused to guide the clinical management of cancer in the presence of microarray data. We propose a strategy based on Bayesian networks to treat clinical and microarray data on an equal footing. The main advantage of this probabilistic model is that it allows to integrate these data sources in several ways and that it allows to investigate and understand the model structure and parameters. Furthermore using the concept of a Markov Blanket we can identify all the variables that shield off the class variable from the influence of the remaining network. Therefore Bayesian networks automatically perform feature selection by identifying the (in)dependency relationships with the class variable.

Results: We evaluated three methods for integrating clinical and microarray data: decision integration, partial integration and full integration and used them to classify publicly available data on breast cancer patients into a poor and a good prognosis group. The partial integration method is most promising and has an independent test set area under the ROC curve of 0.845. After choosing an operating point the classification performance is better than frequently used indices.

Contact: olivier.gevaert@esat.kuleuven.be

1 INTRODUCTION

In the past decade microarrays have had a great impact on cancer research. This technology allows to measure the expression of thousands of genes at once; possibly representing the whole genome. Usually a microarray consists of a selection of probes which are applied onto a solid surface and represent a number of genes (Lockhart *et al*. (1996); Brown and Botstein (1999)). Reverse transcribed mRNA extracted from a tumor sample can be hybridized with the probes on this surface. This results in expression levels of thousands of genes for every tumor sample that is hybridized. The resulting data has been used for many applications such as class discovery and the prediction of diagnosis, prognosis or treatment response. Several studies have been conducted using microarray technology studying several types of cancer (Golub *et al*. (1999); Bhattacharjee *et al*. (2001); Singh *et al*. (2002); van't Veer *et al*. (2002); van de Vijver *et al*. (2002); Spentzos *et al*. (2004, 2005)).

However, microarray data is high dimensional, characterized by many variables and few observations. Moreover this technique suffers from a low signal-to-noise ratio. In our opinion, integration of other sources of information could be important to counter randomly generated differences in expression levels. For example Shedden *et al*. (2003) used a pathological framework and showed that this information significantly lowered the number of genes required in their model. Nevertheless, the focus in most studies is on the microarray analysis while the clinical data is not used in the same manner. Clinical data includes for example: patient history, laboratory analysis or ultrasound parameters. This data was the basis of research and fully guided the clinical management of cancer in the pre-microarray era and is, in our opinion, often underused when microarray data is available. Here we propose methods based on Bayesian networks that integrate clinical data and microarray data. These methods treat both the clinical and the microarray variables (i.e. the gene expression levels) in the same manner. For example, Shedden *et al*. (2003) also did not add clinical data to the gene expression leves when classifying tumour samples.

Bayesian networks are popular decision support models (Husmeier *et al*. (2005)) because they inherently model the uncertainty in the data. They are a successful marriage between probability theory and graph theory. They allow to model a multidimensional probability distribution in a sparse way by searching independency relations in the data. Furthermore this model allows different strategies to integrate two data sources. First, it is possible to combine data sources directly or, secondly, by combining them at the decision level. Furthermore, because Bayesian networks are learned from data in two independent steps, we can define a third method to integrate both data sources. These three methods will be presented and evaluated using Receiver Operator Characteristic (ROC) curves on the training set. The method with the highest average ROC performance will be evaluated on an independent test set. To the author's knowledge, the first two methods have not been previously applied in this context and the third method has not been previously defined.

We will focus as an example on the prediction of the prognosis in lymph node negative breast cancer (without apparent tumor cells

*To whom correspondence should be addressed.

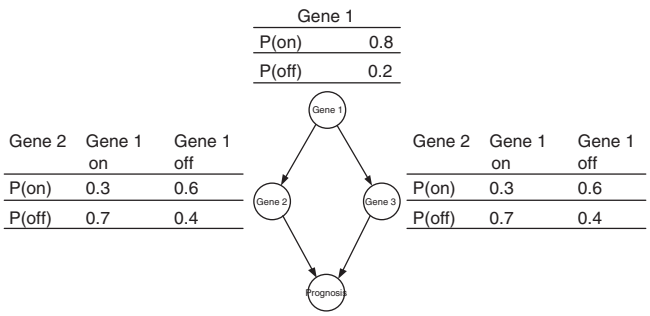


Fig. 1. A simple example of a Bayesian network with four binary variables. The conditional probability tables are shown next to each node where each column in such a table refers to a specific instantiation of the parents. Gene 1 has no parents therefore the node's table specifies a priori probabilities.

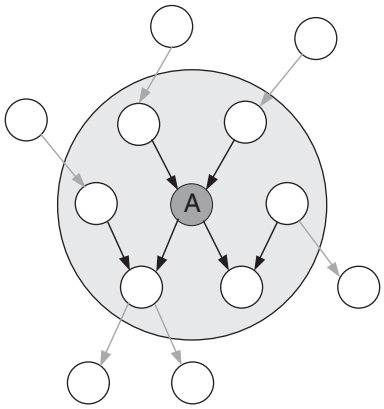


Fig. 2. The Markov blanket of variable A is composed of the variable's parents, its children and its children other parents. Here the Markov blanket variables are shown in a grey circle.

in local lymph nodes at diagnosis). We define the outcome as a variable that can have two values: poor prognosis or good prognosis. Poor prognosis corresponds to recurrence within 5 years after diagnosis and good prognosis corresponds to a disease free interval of at least 5 years (van't Veer *et al*. (2002)). If we can distinguish between these two groups, patients could be treated more optimally thus eliminating over- or under-treatment.

2 METHODS

2.1 Bayesian networks

2.1.1 Definition A Bayesian network is a probabilistic model that consists of two parts: a dependency structure and local probability models (Pearl (1988); Neapolitan (2004)). The dependency structure specifies how the variables are related to each other by drawing directed edges between the variables without creating directed cycles. Each variable depends on a possibly empty set of other variables which are called the parents:

$$p(x_1,\ldots,x_n) = \prod_{i=1}^{n} p(x_i \mid Pa(x_i)) \qquad (1)$$

where $Pa(x_i)$ are the parents of x_i. Usually the number of parents for each variable is small therefore a Bayesian network is a sparse way of writing down a joint probability distribution. The second part of this model, the local probability models, specifies how the variables depend on their parents. We used discrete-valued Bayesian networks which means that these local probability models can be represented with Conditional Probability Tables (CPTs). Such a table specifies the probability that a variable takes a certain value given the value of its parents. Figure 1 shows an example of a Bayesian network with four binary variables. The prognosis variable in this example has two parents: gene 2 and gene 3. The CPTs for each variable are shown alongside each node.

2.1.2 Markov Blanket An important concept of Bayesian networks is the Markov blanket of a variable. The Markov blanket of a variable is the set of variables that completely shields off this variable from the other variables. This set consists off the variable's parents, children and its children's other parents. A variable in a Bayesian network is conditionally independent of the other variables given its Markov Blanket. Conditional independency means that when the Markov blanket of a certain variable x is known, adding knowledge of other variables leaves the probability of x unchanged (Korb and Nicholson (2004)). This is an important concept because the Markov blanket is the only knowledge that is needed to predict the behaviour of that variable. For classification purposes we will focus on the Markov Blanket of the outcome variable. The concept of a Markov blanket is shown in Figure 2.

2.2 Bayesian network learning

Previously we mentioned that a discrete valued Bayesian network consists of two parts. Consequently, there are two steps to be performed during model building: structure learning and learning the parameters of the CPTs.

2.2.1 Structure learning First the structure is learned using a search strategy. Since the number of possible structures increases super-exponentially with the number of variables, we used the well-known greedy search algorithm K2 (Cooper and Herskovits (1992)) in combination with the Bayesian Dirichlet (BD) scoring metric:

$$p(S \mid D) \propto p(S) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left[\frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \right], \qquad (2)$$

with N_{ijk} the number of cases in the data set D having variable i in state k associated with the j-th instantiation of its parents in current structure S. n is the total number of variables. Next, N_{ij} is calculated by summing over all states of a variable: $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. N'_{ijk} and N'_{ij} have similar meanings but refer to prior knowledge for the parameters. When no knowledge is available they are estimated using $N_{ijk} = N/(r_i q_i)$ (Heckerman *et al*. (1995)) with N the equivalent sample size, r_i the number of states of variable i and q_i the number of instantiations of the parents of variable i. $\Gamma(.)$ corresponds to the gamma distribution. Finally $p(S)$ is the prior probability of the structure. $p(S)$ is calculated by: $p(S) = \prod_{i=1}^{n} \prod_{l_i=1}^{p_i} p(l_i \rightarrow x_i) \prod_{m_i=1}^{o_i} p(m_i x_i)$ with p_i the number of parents of variable x_i and o_i all the variables that are not a parent of x_i. Next, $p(a \rightarrow b)$ is the probability that there is an edge from a to b while $p(ab)$ is the inverse, i.e. the probability that there is no edge from a to b. Since we are interested in the prediction of the prognosis, edges with the outcome variable are given a higher prior probability than other edges.

Using Equation 2 we can now score structures using the K2 search strategy. K2 consists of a greedy search combined with a prior ordering of the variables. This ordering restricts the search space by only allowing parents if they precede the current variable in the ordering. Then K2 iteratively tries to find the best parents for each variable separately by starting with an empty set of parents and incrementally adding the best parents. When the addition of a parent does not increase the score, the algorithm stops and moves on to the next variable in the ordering. Since the ordering of the variables is not known in advance, the model building process is iterated a number of times with

different permutations of the ordering. Then the network with the highest score is chosen.

2.2.2 Parameter learning The second step of the model building process consists of estimating the parameters of the local probability models corresponding with the dependency structure. In section 2.1.1 we reported that we are using CPTs to model these local probability models. For each variable and instantiation of its parents there exists a CPT that consists of a set of parameters. Each set of parameters was given a uniform Dirichlet prior:

$$p(\theta_{ij} \mid S) = Dir(\theta_{ij} \mid N'_{ij1}, \ldots, N'_{ijk}, \ldots, N'_{ijr_i}) \quad (3)$$

with θ_{ij} a parameter set where i refers to the variable and j to the j-th instantiation of the parents in the current structure. θ_{ij} contains a probability for every value of the variable x_i given the current instantiation of the parents. *Dir* corresponds to the Dirichlet distribution with $(N'_{ij1}, \ldots, N'_{ijr_i})$ as parameters of this Dirichlet distribution. Parameter learning then consists of updating these Dirichlet priors with data. This is straightforward because the multinomial distribution that is used to model the data, and the Dirichlet distribution that models the prior, are conjugate distributions. This results in a Dirichlet posterior over the parameter set:

$$p(\theta_{ij} \mid D, S) = Dir(\theta_{ij} \mid N'_{ij1} + N_{ij1}, \ldots, N'_{ijk} + N_{ijk}, \ldots, N'_{ijr_i} + N_{ijr_i}) \quad (4)$$

with N_{ijk} defined as before. We summarized this posterior by taking the Maximum A Posteriori (MAP) parameterization of the Dirichlet distribution and used these values to fill in the corresponding CPTs for every variable. Using MCMC could improve our current set-up because this technique allows devising the complete posterior distribution (Neal (1996)).

2.3 Data

We used the data of van't Veer *et al*. (2002) which is available at http://www.rii.com/publications/default.htm or in the Integrated Tumor Transcriptome Array and Clinical data Analysis database (ITTACA (2006)). This data set consists of two groups of patients. The first group of patients, which we call the training set, consists of 78 patients of which 34 patients belonged to the poor prognosis group and 44 patients belonged to the good prognosis group. The second group of patients, the test set, consists of 19 patients of which 12 patients belonged to the poor prognosis group and 7 patients belonged to the good prognosis group. DNA microarray analysis was used to determine the mRNA expression levels of approximately 25000 genes for each patient. Every tumour sample was hybridized against a reference pool made by pooling equal amounts of RNA from each patient. The ratio of the sample and the reference was used as a measure for the expression of the genes and they constitute the microarray data set. Each patient also had the following clinical variables recorded: age, diameter, tumor grade, oestrogen and progesterone receptor status, the presence of angioinvasion and lymphocytic infiltration, which together form the clinical data.

2.3.1 Preprocessing The microarray data consists of approximately 25000 expression values per patient, which was already background corrected, normalized and log-transformed. An initial selection was done (similar to van't Veer *et al*. (2002)) by removing the genes that did not meet the following criteria using only the training data: at least a twofold increase or decrease and a P-value of less than 0.01 in more than 3 tumors. This resulted in a subset of approximately 5000 genes. Then we calculated the correlation between the expression values of these genes with the binary outcome and selected the genes with a correlation of ≥ 0.3 or ≤ -0.3. This resulted in 232 genes that where correlated with the outcome. Missing values were estimated using a 15-weighted nearest neighbours algorithm (Troyanskaya *et al*. (2001)). Then these genes were discretized into three categories: baseline, over-expression or under-expression according to two thresholds. These thresholds depended on the variance of the gene such that a gene with high variance receives a higher threshold than a gene with low variance. The data set that results from these steps was used as input for the Bayesian network software.

2.3.2 Model building We evaluated the performance of the different methods for integrating both data sources (see section 2.4) using the training data. This was done by randomizing the training data set 100 times, in a stratified way, into a set of 70% of the patients used to build the model (model building data set) and a set of 30% to estimate the Area Under the ROC curve (AUC). Then these 100 AUCs were averaged and reported. In this manner we can evaluate the generalizing performance of a specific method and compare with other methods.

Next, the method that performed best in the previous step was used to train 100 models with different orderings using the complete training set. The model with the highest AUC among these 100 models was chosen to predict the outcome on the test set.

2.4 Integration of data sources

2.4.1 Full integration Bayesian networks allow to combine the two data sources, the clinical and microarray data, in different ways. The first method, full integration, is equal to putting both data sources together and treating them as if it is one dataset. This means that both the clinical variables (e.g. age, diameter, grade, etc.) and the microarrays variables (mRNA expressions for each gene) are offered as one data set to the Bayesian network learning algorithm. In this manner the developed model can contain any type of relationship between the clinical variables and the microarray variables.

2.4.2 Decision integration The decision integration method amounts to learning a separate model for the clinical and the microarray data. Then the predictions for the outcome are fused. This comes down to combining the probability of the outcome for the clinical model with the probability of the outcome for the microarray model using weights. The weight parameter is trained using only the model building data set (see section 2.3.2) within each randomization which, in the context of decision integration, is called an outer randomization. This is done by performing again 100 inner randomizations of the model building data set within each outer randomization by again splitting this data set in 70% of the data for training and 30% of the data for testing. For each inner randomization the weight is increased from 0.0 to 1.0 in steps of 0.1. Then the weight value with the highest average AUC on the 30% left out data of the 100 inner randomizations is chosen as weight for the outer randomization.

2.4.3 Partial integration Bayesian networks also allow a third method, which we will call partial integration. This is due to the fact that learning Bayesian networks is a two step process. Therefore we can perform the first step, structure learning, separate for both data sources. This results in a structure for the clinical data and a structure for the microarray data. Both structures have only one variable in common, the outcome, since this variable is present in both data sources. The outcome variable allows joining the separate structures into one structure. Then the second step of learning Bayesian networks (i.e. parameter learning) starts with the combined clinical and microarray data. Partial integration is similar to imposing a restriction during structure learning where no links are allowed between clinical variables and gene expression variables.

3 RESULTS

Model building was done as described in section 2.3.2 for the three integration methods (full, partial and decision integration) and for both data sources (clinical and microarray) separately for comparison. In case of decision integration, we used randomizations to determine the weights to fuse the decisions as described in 2.4.2. This resulted in a weight of 0.6 for predicted probabilities of the clinical model and a weight of 0.4 for predicted probabilities of the microarray model, slightly favouring the clinical model. After choosing these optimal weights, we can compare the methods for integrating the data sources. Table 1 shows the AUCs for the

Table 1. Average AUC performance and standard deviation of the three methods for integrating clinical and microarray data and each data source separately with 100 randomizations. The first two methods, clinical and microarray, are for comparison. The next three methods (decision, partial and full) refer to the methods for integrating the clinical and microarray data.

Method	average AUC	Std
Clinical data	0.751	0.086
Microarray data	0.750	0.073
Decision integration	0.790	0.072
Partial integration	0.793	0.068
Full integration	0.747	0.099

Table 2. The AUC of the Bayesian network models (BPIM and BDIM) and of the reconstructed model based on van't Veer *et al*. 2002 based on 70 genes.

	AUC	std
70 genes	0.851	0.132
BPIM	0.845	0.132
BDIM	0.810	0.118

developed models. Partial integration and decision integration are significantly different from the other methods but not significantly different from each other (Wilcoxon rank sum tests).

Next, both decision integration and partial integration were chosen as the best methods of integrating the two data sources and 100 models were built using the training set. Then the best performing model for each method was chosen and used to predict the outcome on the test data set. The best partial integration model is referred to as BPIM (Best Partial Integration Model) and the best decision integration model as BDIM (Best Decision Integration Model). Table 2 shows the AUC of these two models on the test set. We compared our models with the 70 genes prognosis profile by applying the methods described in van't Veer *et al*. (2002) and using the resulting classifier on the test set. The AUC is also shown in table 2, the standard deviations were estimated according to Hanley and McNeil (1983). Both BPIM and the 70 genes model perform in the same manner on the data set while BDIM is worse. However, there are no significant differences between the ROC curves of BDIM, BPIM and the 70 genes model (Hanley and McNeil (1983)).

Next we chose an operating point for BDIM and BPIM by choosing a threshold that corresponds with a maximum for the sum of the sensitivity and specificity (Smet *et al*. (2004)). Then we compared the classifications of our models with the 70 genes model and with the following indices: the St. Gallen consensus (Goldhirsch *et al*. (1998)), the National Institute of Health (NIH) index (Eifel *et al*. (2001)) and following (Edén *et al*. (2004)) also with the widely used Nottingham Prognostic Index (NPI) (Blamey *et al*. (1979)). For the NPI we used the standard threshold of 3.4 to determine a good or a poor prognosis. Below this threshold the prognosis is considered good, above this threshold the prognosis is considered moderate or poor (Todd *et al*. (1987)). Table 3 shows the number of patients that is assigned to the poor prognosis group for

Table 3. The number of patients assigned a poor prognosis for the complete test set and for the true poor and good prognosis patients.

	Total test set (n = 19)	Metastasis within 5 yr (n = 12)	Disease free at 5 yr (n = 7)
St Gallen 1998[♯]	13/19 (68%)	10/12 (83%)	3/7 (43%)
NIH 2000[°]	15/19 (79%)	10/12 (83%)	5/7 (71%)
NPI[♦]	11/19 (58%)	9/12 (75%)	2/7 (29%)
70 genes[†]	14/19 (74%)	12/12 (100%)	2/7 (29%)
BPIM[†]	13/19 (68%)	11/12 (92%)	2/7 (29%)
BDIM[†]	11/19 (58%)	9/12 (75%)	2/7 (29%)

[♯]Either one of the following criteria equals poor prognosis: ER negative, tumour diameter $\geq$2 cm, grade 3 or age <35

[°]Poor prognosis if tumour diameter >1 cm.

[♦]NPI is the sum of 0.2 times the tumour diameter in cms, lymph node stage and the tumour grade.

[†]The operating point is determined by maximizing the sum of the sensitivity and specificity on the training set.

the complete test set, the set of true poor prognosis patients (i.e. sensitivity) and the set of true good prognosis patients (i.e. 1-specificity). We have applied the St Gallen consensus and the NIH index in the same manner as van't Veer *et al*. (2002). The results show that both the St Gallen consensus and the NIH consensus criteria have a tendency to produce more false positives than the other models which has been observed before (Boyages *et al*. (2002)). In the test set both indices also have some false negatives which can be due to sample selection and small sample size. Both BPIM and the 70 genes have similar performance and are better than the other models since they produce few false positives and false negatives. Both Tables 2 and 3 show that BPIM and the 70 genes have similar performance and are better than BDIM and the frequently used indices. BPIM and 70 genes can reliably be used to predict the prognosis in lymph node negative breast cancer.

Figure 3 shows the complete network built with partial integration. The outcome variable and its Markov Blanket is indicated with triangle nodes. Figure 4 shows the Markov Blanket in detail with the gene names where possible. There are three clinical variables: age, grade and angioinvasion and 13 genes, 12 annotated and 1 unannotated.

4 DISCUSSION

We have developed Bayesian networks to integrate clinical and microarray data using the data of van't Veer *et al*. (2002) and investigated if an improvement was made for the prediction of metastasis in breast cancer. We investigated three methods for integrating the two data sources with Bayesian networks: full integration, partial integration and decision integration.

Table 1 showed that only partial integration and decision integration perform significantly better than each data source separately. We believe that this is due to the different nature of the data sources. Clinical data has a low noise level, there are mostly fewer variables than observations and there are both discrete and continuous-valued variables. Microarray data on the other hand has a much higher noise level. There are a lot more variables than observations and all

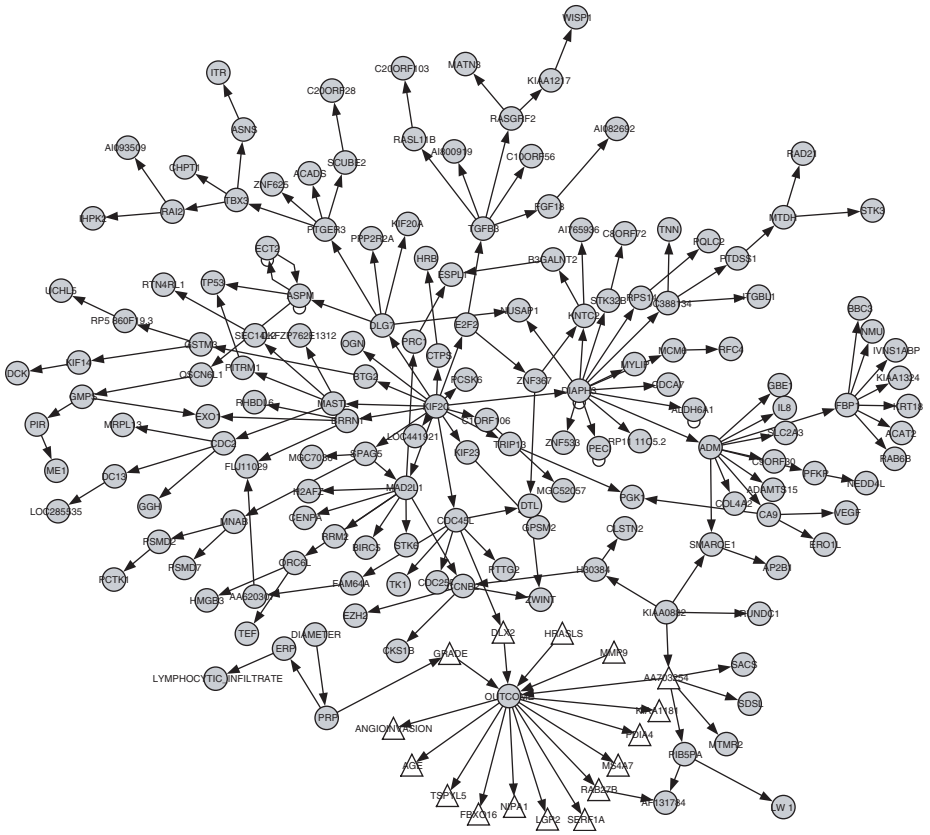


Fig. 3. The complete Bayesian network for the best model using partial integration of clinical and microarray data. The Markov blanket of the outcome variable is indicated with triangle white nodes.

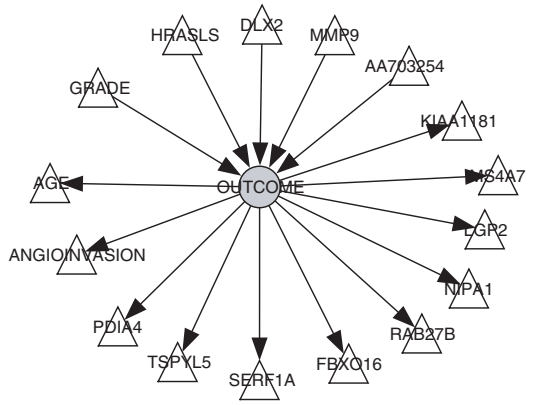


Fig. 4. Markov blanket of the outcome variable for the BPIM model. Gene names have been used where possible.

the variables are continuous. Therefore, it could be advisory to treat them separately in some way. Partial integration uses separate structure learning while decision integration builds separate models but fuses the outcome probabilities. Full integration does not make a distinction between these two heterogeneous data sources which causes that the clinical variables are submerged by the microarray variables and mostly have few connections. This leads to a model where the Markov Blanket only consists of microarray variables and explains the similar performance between full integration and using only the microarray data.

Next, Table 2 showed that BPIM generalizes best to unseen data compared to BDIM. The difference between these two models is that BPIM is integrated at the parameter level and BDIM at the decision level. The former combines clinical and microarray variables in a more sophisticated way because combined parameter learning results in different parameters for every instantiation of the clinical variables. The latter method combines the outcome probabilities using a weighting scheme and relies on the weights for each model. Furthermore BPIM outperforms the prognostic indices and has comparable performance with the 70 genes prognosis profile (van't Veer *et al*. (2002)) despite having fewer genes. This suggests that using clinical data decreases the number of genes required to reliably predict the prognosis. Moreover the low number of genes in BPIM could allow the design of a cheaper test for breast cancer prognosis while still benefiting from data at the molecular level.

Next, we also looked more closely at the BPIM model to investigate the performance of the model when the links of the outcome variable with either the clinical variables or the microarray variables in the Markov blanket are removed. This resulted in worse performance of the model. When the links between the outcome and the clinical variables are removed the AUC performance drops to 0.804 (std 0.130). Similarly when the links between the outcome and the genes are removed the AUC performance drops to

0.798 (std 0.128). This is strong evidence that the combination between the clinical and the microarray variables boosts the performance. Also the formation of a prognostic index from a combination of clinical variables and a small number of genes seems possible.

Furthermore we searched the literature for relations of the variables in the Markov blanket of BPIM (see Figure 4) with breast cancer prognosis and metastasis. The presence of the clinical variables can be explained because they are used as conventional prognostic markers and in prognostic indices. Age in particular because patients with breast cancer at young age have been correlated with poor prognosis (Goldhirsch *et al*. (1998)) while grade is part of the NPI (Blamey *et al*. (1979)). Moreover, recently a large study has shown that lymphovascular invasion, which is related to angioinvasion, is an independent prognostic factor in node-negative breast cancer and improves the NPI (Lee *et al*. (2006)). Furthermore there are 13 genes, 12 annotated and 1 unannotated. Among the annotated genes, MMP9, HRASLS and RAB27B have strong associations with cancer (Owen *et al*. (2004); Kaneda *et al*. (2004)). MMP9 is associated with tumor invasion and angiogenesis since matrix metalloproteases are an important family of proteases that degrade a path through the extra-cellular matrix and the stroma. This process allows tumor cells to invade the surrounding tissue (Pecorino *et al* (2005)). HRASLS is associated with the RAS pathway (Malaney and Daly (2001)) and is thought to function as a tumor suppressor. Furthermore RAB27B is a member of the RAS oncogene family.

On the other hand BDIM also showed interesting characteristics. This decision integration model used a weight of 0.6 for the clinical model and a weight of 0.4 for the microarray model. This emphasizes the importance of the clinical data for classification compared to the microarray data. In addition, the clinical data generalizes better to new data since the test set performance is similar to the training set performance (average training set AUC of 100 clinical data models is 0.838) while the microarray data allows bitter fitting but with the danger of overfitting (average training set AUC of 100 microarray data models is 0.981) (also see Table 1). Therefore combining both data sources can lead to models benefiting from the complementary advantages of each data source separately. The results of BDIM and BPIM show that this is possible.

The advantages of the probabilistic approach are that the current models can be extended with prior information. This can be done both at the structure level and the parameter level. This will influence the variables that show up in the Markov blanket and results in a feature selection method based on data and prior biological knowledge with automatic tuning of the balance between data and prior knowledge. Possible sources of prior information are literature abstracts (Glenisson *et al*. (2004)), known pathways (e.g. KEGG or BIOCARTA) or motif information (Thijs *et al*. (2002)). Moreover publicly available microarray data sets studying the same clinical problem can be combined via the prior.

Furthermore, since Bayesian networks are not tuned for classification—they provide a more general framework by modeling a multi-dimensional probability distribution—the reported performance could be improved by using more traditional classifiers. Our ongoing research includes investigating the use of Bayesian networks as feature selector followed by Least Squares Support-Vector Machines for classification (Pochet *et al*. (2004)).

In conclusion, the integrated use of clinical and microarray data outperforms the indices based on clinical data (NIH, St. Gallen and NPI) and has comparable performance with the 70 genes prognosis profile. Therefore this approach offers possibilities for the use of Bayesian networks to integrate data sources for other types of cancer and data. Furthermore BPIM has comparable performance as the 70 genes prognosis profile (van't Veer *et al*. (2002)) but allows interpretation and contains fewer genes. When more public data becomes available the described approach and BPIM in particular can be validated.

ACKNOWLEDGEMENTS

This research is supported by: the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). Research Council KUL: GOA AMBioRICS, CoE EF/05/007 SymBioSys, IDO (Genetic networks), several PhD/postdoc and fellow grants Flemish Government: FWO PhD/postdoc grants, projects G.0407.02 (support vector machines), G.0413.03 (inference inbioi), G.0388.03 (microarrays for clinical use), G.0229.03 (ontologies in bioi), G.0241.04 (Functional Genomics), G.0499.04 (Statistics), G.0232.05 (Cardiovascular), G.0318.05 (subfunctionalization), G.0553.06 (VitamineD), research communities (ICCoS, ANMMM, MLDM); IWT PhD Grants, GBOU-McKnow (Knowledge management algorithms), GBOU-SQUAD (quorum sensing), GBOU-ANA (biosensors), TAD-BioScope, Silicos. Belgian Federal Science Policy Office: IUAP P5/22 ('Dynamical Systems and Control: Computation, Identification and Modelling, 2002-2006); EU-RTD: FP5-CAGE (Compendium of Arabidopsis Gene Expression); ERNSI: European Research Network on System Identification; FP6-NoE Biopattern; FP6-IP e-Tumours, FP6-MC-EST Bioptrain.

REFERENCES

Bhattacharjee,A., Richards,W., Staunton,J., Li,C., Monti,S., Vasa,P., Ladd,C., Beheshti,J., Bueno,R., Gillette,M., Loda,M., Weber,G., Mark,E., Lander,E., Wong,W., Johnson,B., Golub,T., Sugarbaker,D. and Meyerson,M. (2001) Classification of human lung carcinomas by mrna expression profiling reveals distinct adenoma subclasses. *PNAS*, **98**, 13790–5.

Blamey,R., Davies,C., Elston,C., Johnson,J., Haybittle,J. and Maynard,P. (1979) Prognostic factors in breast cancer—the formation of a prognostic index. *Clin Oncol*, **5**, 227–236.

Boyages,J., Chua,B., Taylor,R., Bilous,M., Salisbury,E., Wilcken,N. and Ung,O. (2002) Use of the st gallen classification for patietns with node-negative breast cancer may lead to overuse of adjuvant chemotherapy. *British journal of surgery*, **89**, 789–796.

Brown,P. and Botstein,D. (1999) Exploring the new world of the genome with dna microarrays. *Nature*, **21**, 33–7.

Cooper,G. and Herskovits,E. (1992) A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, **9**, 309–347.

Edén,P., Ritz,C., Rose,C., Fernö,M. and Peterson,C. (2004) Good old clinical markers have similar power in breast cancer prognosis as microarray gene expression profilers. *European journal of cancer*, **40**, 1837–1841.

Eifel,P., Axelson,J., Costa,J. and *et al*. (2001) National institutes of health consensus development conference statement: adjuvant therapy for breast cancer. *J Natl Cancer Inst*, **93**, 979–989.

Glenisson,P., Coessens,B., Vooren,S.V., Mathys,J., Moreau,Y. and Moor,B.D. (2004) Txtgate: profiling gene groups with text-based information. *Genome Biology*, 5.

Goldhirsch,A., Glick,J., Gelber,R. and Senn,H. (1998) Meeting highlights: international consenus panel on the treatment of primary cancer. *J Natl Cancer Inst*, **90**, 1601–1608.

Golub,T., Slonim,D., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J., Coller,H., Loh,M., Downing,J., Caligiuri,M., Bloomfield,C. and Lander,E. (1999) Molecular

classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–7.

Hanley,J. and McNeil,B. (1983) A method of comparing the areas under receiver operating characteristics curves derived from the same cases. *Radiology*, **148**, 839–43.

Heckerman,D., Geiger,D. and Chickering,D. (1995) Learning bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, **20**, 197–243.

Husmeier,D., Dybowski,R. and Roberts,S., eds (2005) *Probabilistic modelling in bioinformatics and medical informatics*. Springer-Verlag, London, UK.

ITTACA (2006). http://bioinfo-out.curie.fr/ittaca.

Kaneda,A., Wakazono,K., Tsukamoto,T., Watanabe,N., Yagi,Y., Tatematsu,M., Kaminishi,M., Sugimura,T. and Ushijima,T. (2004) Lysyl oxidase is a tumor suppressor gene inactivated by methylation and loss of heterozygosity in human gastric cancers. *Cancer Res*., **64**, 6410–6415.

Korb,K. and Nicholson,A. (2004) *Bayesian artificial intelligence*. Chapman and Hall, Boca Raton, Florida.

Lee,A., Pinder,S., Macmillan,R., Mitchell,M., Ellis,I., Elston,C. and Blamey,R. (2006) Prognostic value of lymphovascular invasion in women with lymph node negative invasive breast carcinoma. *European journal of cancer*, **42**, 357–362.

Lockhart,D., Dong,H., Byrne,M., Follettie,M., Gallo,M., Chee,M., Mittman,M., Wang,C., Kobayashi,M., Horton,H. and Brown,E. (1996) Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, **14**, 1675–80.

Malaney,S. and Daly,R. (2001) The ras signaling pathway in mammary tumorigenesis and metastasis. *J Mammary Gland Biol Neoplasia*, **6**, 101–113.

Neal,R. (1996) *Bayesian learning for neural networks*. Springer-Verlag, New York.

Neapolitan,R. (2004) *Learning Bayesian networks*. Prentice Hall, Upper Saddle River, NJ.

Owen,J., Iragavarapu-Charyulu,V. and Lopez,D. (2004) T cell-derived matrix metalloproteinase-9 in breast cancer: friend or foe? *Breast Dis*, **20**, 145–153.

Pearl,J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Matteo, California.

Pecorino,L. (2005) *Molecular biology of cancer*. Oxford university press, New York.

Pochet,N., Smet,F.D., Suykens,J. and Moor,B.D. (2004) Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. *Bioinformatics*, **20**, 3185–95.

Shedden,K., Taylor,J., Giordano,T., Kuick,R., Misek,D., Rennert,G., Schwartz,D., Gruber,S., Logsdon,C., Simeone,D., Kardia,S., Greenson,J., Cho,K., Beer,D., Fearon,E. and Hanash,S. (2003) Accurate molecular classification of human cancers based on gene expression using a simple classifier with a pathological tree-based framework. *American journal of pathology*, **163**, 1985–1995.

Singh,D., Febbo,P., Ross,K., Jackson,D., Manola,J., Ladd,C., Tamayo,P., Renshaw,A., D'Amico,A., Richie,J., Lander,E., Loda,M., Kantoff,P., Golub,T. and Sellers,W. (2002) Gene expression correlates of clinical prostate cancer behaviour. *Cancer Cell*, **1**, 203–9.

Smet,F.D., Moreau,Y., Engelen,K., Timmerman,D., Vergote,I. and Moor,B.D. (2004) Balancing false positives and false negatives for the detection of differential expression in malignancies. *British Journal of Cancer*, **91**, 1160–1165.

Spentzos,D., Levine,D., Kolia,S., H,O., Boyd,J., Libermann,T. and Cannistra,S. (2005) Unique gene expression profile based on pathologic response in epithelial ovarian cancer. *J Clin Oncol*, **23**, 7911–8.

Spentzos,D., Levine,D., Ramoni,M., Joseph,M., Gu,X., Boyd,J., Libermann,T. and Cannistra,S. (2004) Gene expression signature with independent prognostic significance in epithelial ovarian cancer. *J. Clin. Oncol*., **22**, 4700–10.

Thijs,G., Moreau,Y., Smet,F.D., Mathys,J., Lescot,M., Rombauts,S., Rouze,P., B,B.D.M. and Marchal,K. (2002) Inclusive: integrated clustering, upstream sequence retrieval and motif sampling. *Bioinformatics*, **18**, 331–2.

Todd,J., Dowle,C., Williams,M., Elston,C., Ellis,I., Hinton,C., Blamey,R. and Haybittle,J. (1987) Confirmation of a prognostic index in primary breast cancer. *Br J Cancer*, **56**, 489–492.

Troyanskaya,O., Cantor,M., Sherlock,G., Brown,P., Hastie,T., Tibshirani,R., Botstein,D. and Altman,R. (2001) Missing value estimation methods for dna microarrays. *Bioinformatics*, **17**, 520–525.

van de Vijver,M., He,Y., van't Veer,L., Dai,H., Hart,A., Voskuil,D., Schreiber,G., Peterse,J., Roberts,C., Marton,M., Parrish,M., Atsma,D., Witteveen,A., Glas,A., Delahaye,L., van der Velde,T., Bartelink,H., Rodenhuis,S., Rutgers,E., Friend,S. and Bernards,R. (2002) A gene expression signature as a predictor of survival in breast cancer. *The new England journal of medicine*, **347**, 1999–2009.

van't Veer,L., Dai,H., van de Vijver,M., He,U., Hart,A., Mao,M., Peterse,H., van der Kooy,K., Marton,M., Witteveen,A., Schreiber,G., Kerkhoven,R., Roberts,C., Linsley,P., Bernards,R. and Friend,S. (2002) Gene expression profiling predicts clinical outcome in breast cancer. *Nature*, **415**, 530–536.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e191–e196
doi:10.1093/bioinformatics/btl206

ZPRED: Predicting the distance to the membrane center for residues in α-helical membrane proteins

Erik Granseth[1], Håkan Viklund[1] and Arne Elofsson[1,*]

[1]Center for Biomembrane Research, Stockholm University, SE-106 91 Stockholm, Sweden

ABSTRACT

Motivation: Prediction methods are of great importance for membrane proteins as experimental information is harder to obtain than for globular proteins. As more membrane protein structures are solved it is clear that topology information only provides a simplified picture of a membrane protein.

Here, we describe a novel challenge for the prediction of α-helical membrane proteins: to predict the distance between a residue and the center of the membrane, a measure we define as the *Z*–coordinate.

Even though the traditional way of depicting membrane protein topology is useful, it is advantageous to have a measure that is based on a more "physical" property such as the *Z*–coordinate, since it implicitly contains information about re-entrant helices, interfacial helices, the tilt of a transmembrane helix and loop lengths.

Results: We show that the *Z*–coordinate can be predicted using either artificial neural networks, hidden Markov models or combinations of both. The best method, ZPRED, uses the output from a hidden Markov model together with a neural network. The average error of ZPRED is 2.55Å and 68.6% of the residues are predicted within 3Å of the target *Z*–coordinate in the 5–25Å region. ZPRED is also able to predict the maximum protrusion of a loop to within 3Å for 78% of the loops in the dataset.

Availability: Supplementary information and training data is available at http://www.sbc.su.se/~erikgr/

Contact: arne@bioinfo.se

1 INTRODUCTION

Integral α-helical membrane proteins constitute an important subset of the proteins encoded by a genome, comprising 20–25% of the proteome (Krogh *et al*., 2001; Granseth *et al*., 2005a). These proteins are crucial for many cellular processes including signaling and transport processes. They are also the target for the majority of all drugs, making them important for the pharmacological industry (Chen and Rost, 2002). For several experimental reasons it is more difficult to obtain the structures of transmembrane proteins than those of globular proteins and a consequence of this is that less than 1% of the 3D-structures in the Protein Data Bank are from transmembrane proteins (Berman *et al*., 2000). Nevertheless it has recently been noted that the number of experimentally known 3D-structures has an exponential increase (White, 2004). Still, for many membrane proteins only "low-resolution" topology information about the structure is known, i.e. what parts of the sequence are transmembrane regions and the orientation of the protein relative to the membrane.

Partly due to the lack of three-dimensional information of membrane proteins many topology predictors have been developed for α-helical transmembrane (TM) proteins. The first only relied on the fact that TM helices are on average more hydrophobic than the loop regions and globular proteins and classified each segment that was sufficiently long and hydrophobic as a TM helix (von Heijne, 1992). Although these simple methods worked surprisingly well, many regions were wrongly classified. A significant improvement was obtained when hidden Markov models (HMMTOP (Tusnády and Simon, 1998), TMHMM (Sonnhammer *et al*., 1998)) were used to extract the features of different regions in TM proteins. Several recent benchmarks have shown that the state of the art methods perform quite well (Chen *et al*., 2002; Käll and Sonnhammer, 2002), predicting the correct topology for close to 70% of the membrane proteins.

For a long time the general view was that membrane proteins in principle existed in a two–dimensional space, with the TM helices perpendicularly penetrating the membrane (Taylor *et al*., 1994). However, recent analysis of membrane protein structures shows that membrane proteins certainly not can be seen as constrained in two dimensions (Granseth *et al*., 2005b). Instead it is clear that many membrane proteins have a similar amount of structural complexity as globular proteins. This can be illustrated by the structure of the glutamate transporter homolog from *Pyrococcus horikoshii* (Yernool *et al*., 2004), Figure 1a. The structure does not only have ordinary TM helices but also two helices that are not helical throughout the entire membrane, one of them contains a helix inside the lipid bilayer that is parallel to the membrane plane. This can also be seen in the corresponding *Z*–coordinate (the distance to the center of the membrane for each residue) located around residue number 140 in Figure 1b. The structure also contains two re-entrant helices, where a helix only goes half-way through the membrane, and then turns back again to the same side it originated from. These two re-entrant helices meet each other in the middle of the membrane, a feature that also can be observed in aquaporin-like structures (Tornroth-Horsefield *et al*., 2006).

Here, we introduce a novel challenge for structure prediction of membrane proteins: the prediction of the *Z*–coordinate, i.e. the distance for a residue to the center of the membrane. Even though the traditional way of depicting membrane protein topology is useful, it is advantageous to also have a measure that is based on a more "physical" property such as the *Z*–coordinate. The problem should

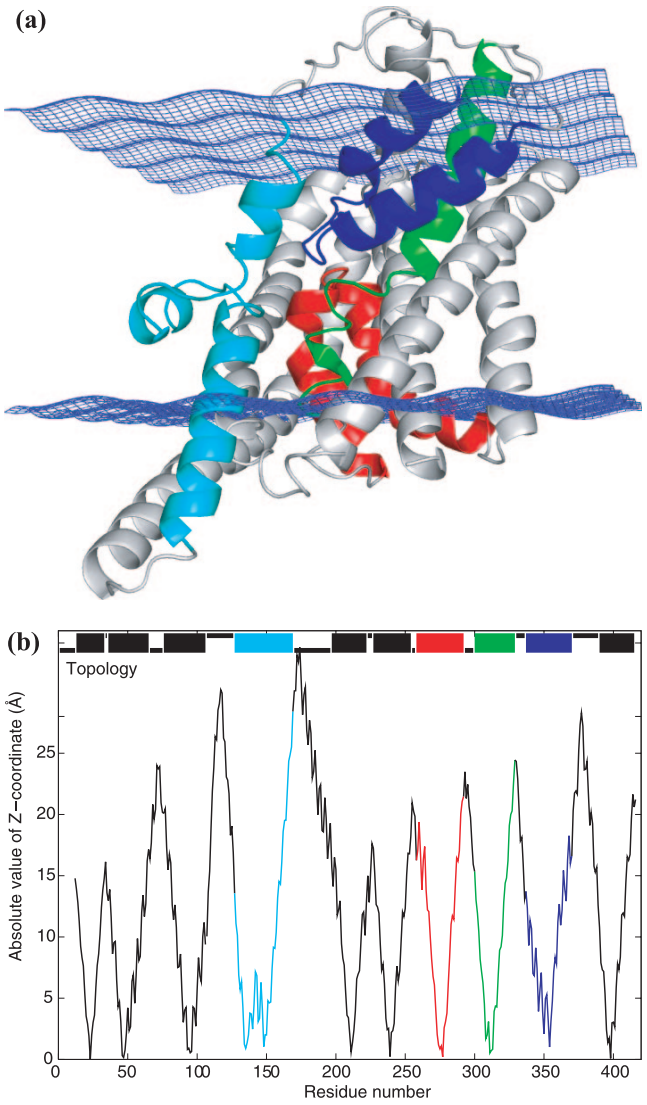


Fig. 1. **(a)** The glutamate transporter homolog (1XFH) contains reentrant regions and non-ideal transmembrane helices. Non-ideal TM helices light-blue and green, reentrant regions red and dark-blue. The end of the hydrocarbon region of the lipid bilayer at ±15Å is depicted as blue mesh. **(b)** The upper part of the image shows the topology of the membrane protein structure. BlackSquares are TM helices, black lines below the midpoint are inside loops, and outside loops if above the midpoint. The lower part shows the absolute value of the *Z*–coordinate. The coloring is the same as in Figure 1a.

not only be seen as an intermediate approach towards complete 3D-structure prediction, but also be of potential interest for the identification of interesting structural features important for drug-binding and/or function of membrane proteins. One such example is locating good *N*-glycosylation sites for experimental topology determination since efficient *N*-glycosylation requires that the acceptor site is sufficiently spaced from the membrane surface (Popov *et al*., 1997).

We have developed a number of methods based on earlier topology predictors to predict the *Z*–coordinate. We show that methods using either hidden Markov models (HMM) or artificial neural networks (ANN) are able to predict the *Z*–coordinate in the 5–25Å region with an average error of ∼3Å while methods that combine both HMMs and ANNs can decrease the average error to 2.55Å. As expected, the use of evolutionary information also provides improvement of the predictions. We show that these predictors can provide valuable additional information complementary to the predictions from traditional topology modeling methods.

2 RESULTS AND DISCUSSION

Topology prediction of membrane proteins has been a valuable tool for classification of membrane proteins, for genomic studies and as an aid for remote homology detection (Hedman *et al*., 2002). However, given the observation that membrane protein structures are complex, the structural information obtained from topology predictions might be limited. In contrast, the *Z*–coordinates of the amino acids in the structure implicitly contain information about re-entrant helices, interfacial helices, the tilt of a TM helix and how much a loop protrudes from the membrane.

A significant difference between our *Z*–coordinate predictor and a topology predictor is that the former only predicts the distance from the center of the membrane and not the direction of this distance, i.e. we do not distinguish between the cytosolic ("inside") and extracellular ("outside") sides of the membrane. This somehow simplifies the problem but is possible since the membrane to a large degree is symmetric. A further simplification used in this study is that all residues that are between 0–5Å are defined to be in a central hydrophobic region and hence set to 5Å. All residues that are outside the lipid bilayer, >25Å are in a similar manner defined to be in a non-membrane environment and set to 25Å. This means that the predictor focuses the predictions on the region where the environment inside the membrane changes most (White and Wimley, 1999). Using a larger region for the prediction (0–30Å instead of 5–25Å) decreased the overall prediction accuracy as well as the accuracy in the 5–25Å region. We also made attempts at using a non-symmetric definition of the membrane, i.e. predicting the −25 to +25Å region, but the prediction accuracy that was obtained for this problem was very low.

Below, we will first describe a set of predictors based on ANNs and/or HMMs that all predict the *Z*–coordinate and thereafter we exemplify the usefulness with the prediction of the glutamate receptor homolog shown in Figure 1.

Prediction accuracy

To be able to assess the quality of the predictions three measures of accuracy are introduced, the average error, the fraction of residues with an error smaller than 3Å and ZQ2, i.e. the fraction of residues correctly predicted to be inside ($\leq$15Å) or outside ($\geq$15Å) the membrane. The ZQ2 resembles Q2 which is often used for benchmarking topology prediction methods. Although these three measures are strongly correlated they provide slightly different types of information as can be seen below.

Prediction of the Z–coordinates using HMM-based methods The standard TMHMM-2.0 model (Krogh *et al*., 2001) predicts the *Z*–coordinate with an average error of 3.17Å, Table 1. Interestingly, the average error did not decrease if evolutionary profiles were used as in PRO-TMHMM, however the amount of residues within 3Å from the target *Z*–coordinate increased slightly. This contradiction is because TMHMM-2.0 frequently uses separate state compart-

ments for short and long globular loops, while PRO-TMHMM uses the same. This leads to a decrease in the *Z*–coordinate accuracy in the loop regions for PRO-TMHMM since it is less specific. Since PRO-TMHMM is superior to TMHMM-2.0 at topology predictions (Viklund and Elofsson, 2004), it has more residues predicted within 3Å.

The HMM based method with best *Z*–coordinate accuracy is PRODIV-TMHMM with an average error of 2.83Å. It predicts 65.9% of the residues within 3Å from their target *Z*–coordinate and has a ZQ2 value of 86.5%. PRODIV-TMHMM differs from PRO-TMHMM by using the target profile to re-estimate the model parameters. This procedure provides a topology prediction which maximizes the divergence of the amino acid distributions in the different regions, something that has been shown to significantly improve the accuracy of topology predictions (Tusnády and Simon, 2001). The improved topology predictions result in improved *Z*–coordinate predictions.

It can be noted that the hidden Markov models have particular problems at predicting the 5–15 Å region, Figure 2. This is largely due to the model architecture of the membrane spanning regions, which contains an intrinsic contradiction between the length variability of the membrane regions and accurate modeling of their *Z*–coordinates. We believe that it should be possible to improve the *Z*–coordinate predictions by using a more sophisticated HMM architecture.

Prediction of the Z–coordinate using artifical neural network based methods A simple neural network trained on the sequence using sparse encoding of a sequence window as input (ZPRED-SEQ) does not outperform any of the HMM based methods, Table 1. However, it performs better than the simplest method assigning the *Z*–coordinate based on the average hydrophobicity alone. The output from the sequence network is very noisy, and it often mispredicts parts of 25Å regions to be below 20Å. This means that the network cannot discriminate between short hydrophobic regions in cytoplasmic or periplasmic domains and longer hydrophobic transmembrane regions. It is not until ZPRED-SEQ's window size is larger than 9 residues that it outperforms the hydrophobicity (data not shown). It is interesting to see that it is possible to predict the *Z*–coordinate, albeit with quite poor accuracy, by the sequence alone. This implies that it is the local environment surrounding a residue that, to a large extent, determines its depth inside the membrane.

The use of evolutionary profiles (ZPRED-PRO) improves the performance, the average error decreases ~0.5Å, the residues within 3Å increase by 15.5% and the ZQ2 value improves slightly compared to ZPRED-SEQ. This improvement is quite dramatic and a notable difference is that when using the evolutionary profiles the network predicts a significant number of residues to be at 5 or 25Å which ZPRED-SEQ rarely does.

In contrast to the HMMs, the error of the ANNs is largest around 15Å and at the two extreme points, Figure 2. The large central error is quite likely due to that it is "easier" to make a mistake in this region since you can both predict a too high and a too low number while at the end you can only make the mistakes in one direction. The increase in the errors at 5 and 25Å are most likely due to that the ANN has not converged completely and should be possible to overcome using more data and improved methods in the future.

Combinations of neural networks and hidden Markov models There are two different possibilities to combine the methods: either by using previously trained neural networks as additional input into the hidden Markov model or by using the output from a hidden Markov model as input to a neural network.

It is not obvious how to include the predicted *Z*–coordinate directly into the HMMs. Therefore a special version of neural networks (ZPRED-D) was trained to predict discrete regions of the *Z*–coordinates. These predictions were then used as an additional alphabet in the HMMs, see methods for details. The inclusion of neural network predictions into the HMMs only lead to significant improvements for TMHMM-2.0 and PRO-TMHMM but not for PRODIV-TMHMM, Table 1. There exist two explanations for this: first, it is harder to improve PRODIV-TMHMM since its original topology prediction performance is better, and second, the parameter re-estimation step of PRODIV-TMHMM restricts the possibility to make small adjustments in the prediction since each state is more optimized to emit a specific amino acid distribution corresponding to a particular sequence position. Interestingly, if the results from the discrete network predictions are included in TMHMM-2.0, the accuracy in the 5–17Å region increases, while it at the same time decreases in the 17–25Å region (data not shown).

The inclusion of the HMM information into the neural networks is straightforward as the output from the HMM can be used as an additional input. Using the output from PRODIV-TMHMM and evolutionary profiles as input to a neural network (ZPRED) produced the method of choice for predicting the *Z*–coordinate, see Table 1. The average error was 2.55Å, around one half turn of a transmembrane helix, and more than two-thirds of the residues were predicted to be within 3Å from the target *Z*–coordinate. The accuracy in the 5–25Å region clearly increased, particularly around 10–20Å, see Figure 2. The accuracy is also improved in the 5–6Å region due to the fact that the neural network without HMM input (ZPRED-PRO) sometimes has problems reaching 5Å, the predictions are often around 5.5–6Å instead. The same tendency could also be seen at the ⩾25Å region. ZPRED also has a more flat distribution of the average error across the 5–25Å region than the other methods.

The glutamate transporter homolog from *Pyrococcus horikoshii*

To illustrate the prediction of the *Z*–coordinate for a complex membrane protein, we studied the prediction of the glutamate transporter homolog in more detail.

Figure 3a shows all residues that are predicted to be ≤15Å by ZPRED. It can be noted that most helices are identified correctly, but that some regions (at the left in this figure) are misplaced. When using a standard TMHMM prediction, slightly larger areas were missed (data not shown).

As described previously, the glutamate transporter homolog contains two re-entrant regions. The predicted *Z*–coordinate shown in Figure 3b contains some indication that there is something peculiar happening close to the first re-entrant helix (residue 270). However, the predicted *Z*–coordinate is located around 10Å instead of 5Å. The second reentrant helix (at residue 350) has a *Z*–coordinate similar to an ordinary TM helix, i.e. it is not identified. This is most likely because it is more hydrophobic than the first one.

Table 1. Performance of predicting the *Z*–coordinate with different methods.

Method	Average error (Å)	Residues within 3Å (%)	ZQ2 (%)
Hydrophobicity	4.26	39.0	78.9
HMM based methods			
TMHMM-2.0	3.17	61.5	84.0
PRO-TMHMM	3.25	62.4	84.0
PRODIV-TMHMM	2.83	65.9	86.5
ANN based methods			
ZPRED-SEQ	3.53	54.1	83.4
ZPRED-PRO	3.01	62.5	86.6
Combined methods			
TMHMM-2.0+ZPRED-D	2.85	64.6	85.1
PRO-TMHMM+ZPRED-D	2.98	64.8	85.7
PRODIV-TMHMM+ZPRED-D	2.78	66.6	86.8
ZPRED	2.55	68.6	87.9

The average error is how far on average each residue deviates from its target *Z*–coordinate. Residues within 3Å is the fraction of residues with an error smaller than 3Å. The ZQ2 score is the fraction of the residues that are correctly predicted to be within 15Å from the center, i.e. inside the membrane, plus the fraction correctly predicted to be outside the membrane.

The fourth TM helix (at residue 140) is the odd one containing the parallel helix in the middle of the membrane. The *Z*–coordinate of this parallel helix is erroneously predicted to be around 15Å, when it should be between 5 and 7Å, but at least it could be possible to identify it. The complexity of this region is completely missed by the PRODIV-TMHMM prediction, which assigns the most hydrophobic region to be the TM helix. In fact, the beginning of the TM helix and the parallel middle helix have similar hydrophobicity as an ordinary loop.

Possibly the most important observation from the *Z*–coordinate prediction of the glutamate transporter homolog and from most other proteins is that the *Z*–coordinates of loop regions can be predicted quite accurately. ZPRED is able to predict the maximum protrusion of a loop to within 1Å for 50% of the loops in the dataset, and within 3Å for 78%. The loops surrounding the re-entrant regions are less accurately predicted.

Topology prediction

Finally, we wanted to estimate if the *Z*–coordinate predictions could be used to improve topology predictions. A simple method to obtain a topology prediction is by assuming all regions that are below a specific *Z*–coordinate to belong to TM helices and choosing inside/outside loop with the positive inside rule (see methods for details). A previously used dataset of 147 experimentally verified membrane protein topologies was used here (Viklund and Elofsson, 2004). Using this strategy ZPRED-SEQ correctly predicts the topology of 40 proteins and ZPRED-PRO 65 proteins, with the latter slightly better than TMHMM-2.0 (61). By including the prediction from PRODIV-TMHMM into ZPRED the topology is correctly predicted 95 times which is 2 fewer than the initial PRODIV-TMHMM predictions. When output from discrete network ZPRED-D was used in the PRO-TMHMM model, a modest improvement was observed (95 vs. 90) while no improvement was observed for PRODIV-TMHMM.

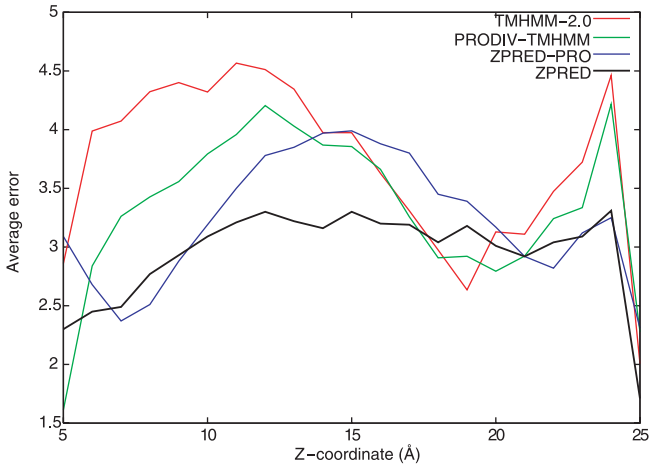


Fig. 2. Average error in the 5–25Å region for different *Z*–coordinate prediction methods.

3 CONCLUSIONS

In this study we have shown that the distance to the center of the membrane can be predicted with acceptable accuracy for residues in α-helical membrane proteins. The prediction can be performed using either an artificial neural network or a hidden Markov model with roughly the same error rate. It was clear that the local environment around a residue to a large degree determines its depth inside the membrane as it was possible to predict the *Z*–coordinate using only hydrophobicity or local sequence information.

The best predictions were obtained using the output from a hidden Markov model as an additional input to an artificial neural network together with profile information from a sequence window. This method, ZPRED, reached an average error of 2.55Å. ZPRED also has a quite evenly distributed average error in the 5–25Å region in contrast to the other methods examined.

While introducing the field of *Z*–coordinate prediction it is also our hope that it will provide an interesting challenge for other developers and that more refined methods will become available as the amount of solved 3D–structures increases. We foresee several possible improvements in the future, for instance, it is clear that the model architecture of TMHMM is not ideal for the predictions of *Z*–coordinates. Hence, we expect that it is possible to improve HMM-based predictions by refining the model to better suit *Z*–coordinate prediction.

4 MATERIAL AND METHODS

Dataset

The dataset consisted of 101 non-homologous protein chains from 46 PDB structures obtained by X-ray diffraction (see supplementary information for full list). The biological unit PDB structures were rotated and translated as described in Tusnády *et al.*, 2005 so that they are positioned in their most

probable localization in the lipid bilayer. The Z–coordinate is then perpendicular to the membrane plane and Z = 0 is in the middle of the membrane. Some of the structures were translated a few Å along the Z–coordinate to better fit their hydrophobicity profiles. In all, 21,589 residues and their corresponding Z–coordinates were used for training and testings and hidden Markov models.

In order to maximize the amount of data, we used the absolute value of the Z–coordinate from the structure and limited it so that all residues above 25 were set to 25Å and all residues between 0 and 5 were set to 5Å. This target value was used for the training and testing of the neural networks and used to benchmark the different hidden Markov models. We also tested a larger region, 0–30Å, but this seriously decreased the learning capabilities of the neural networks. When trying to predict from −25 to +25Å instead, the average error was 13Å and only 20% of the residues were predicted within 3Å of the target Z–coordinate.

147 membrane protein sequences with experimentally verified topologies were used for evaluating the topology prediction (Viklund and Elofsson, 2004).

Hydrophobicity

The hydrophobicity was calculated using the GES scale from Engelman *et al*., 1986 and a running average over 19 residues. The hydrophobicity was extrapolated to the Z–coordinate by linear regression. If the extrapolated hydrophobicity was above 25 it was set to 25Å, and if below 5, set to 5Å.

Neural network training

For the sequence-only neural network (ZPRED-SEQ), the amino acids were converted to numerical values by sparse encoding. PSI-BLAST was used to generate profiles for the profile networks (ZPRED-PRO and ZPRED) (Altschul *et al*., 1997). The log-odds profile from the first iteration was used and converted to values between 0 and 1 by the logistic function $1/(1+e^{-x})$.

The neural networks were 5-fold cross validated, where 4 sets were used for training and the fifth used for testing. All values reported are from the test set data. Netlab (Bishop, 1995) was used for constructing one hidden layer, feed-forward, back propagation networks with linear output nodes and scaled conjugate optimization as optimization algorithm.

The input for the neural networks was a symmetrical sliding window between 3 and 35 residues wide and the target Z–coordinate was for the residue in the middle of the window. Starting with 5 hidden nodes, the average error between the predicted Z–coordinate and the target Z–coordinate stopped decreasing after a window size of 19 residues. Increasing the number of hidden nodes did not increase performance, while a decrease to 4 nodes did not alter the accuracy at all, but has the advantage of decreasing the number of free variables to optimize. 3 nodes seriously decreased the performance, so the final networks used 19 residue sliding windows and 4 hidden nodes.

The learning rate was varied, but for the final networks a learning rate of 0.01 was used. The learning was stopped when the average error ceased to decrease for the test set data.

For the topology prediction of the 147 membrane proteins, the arithmetic average was used from the outputs of the final 5 cross validated networks.

6 different neural networks (ZPRED-D) were trained to mimic the time-averaged distributions of the principal (quasi-molecular) structural groups of a dioleoylphosphocholine (DOPC) bilayer (White and Wimley, 1999). The different regions are: CH_3 (0–5Å), hydrocarbon core (0–15Å), C=C (5–15Å), carbonyl (12–18Å), cholin (17–25Å) and water of hydration (20–25Å). These particular intervals were chosen because they might have specific amino acid composition signatures. A 19 residue sliding window was used to train each of the 6 different networks with logistic output nodes and one node in the hidden layer. The target value was set to 1 in the specific regions and 0 elsewhere. The 6 different networks were 5-fold cross validated and the Mathews Correlation Coefficient (MCC) was used to measure the performance. The final 0–5Å network had MCC 0.53, 0–15Å 0.68, 5–15Å 0.43, 12–18Å 0.15, 17–25Å 0.66 and 20–25Å 0.65. The outputs from these networks were later used as input to the hidden Markov models.

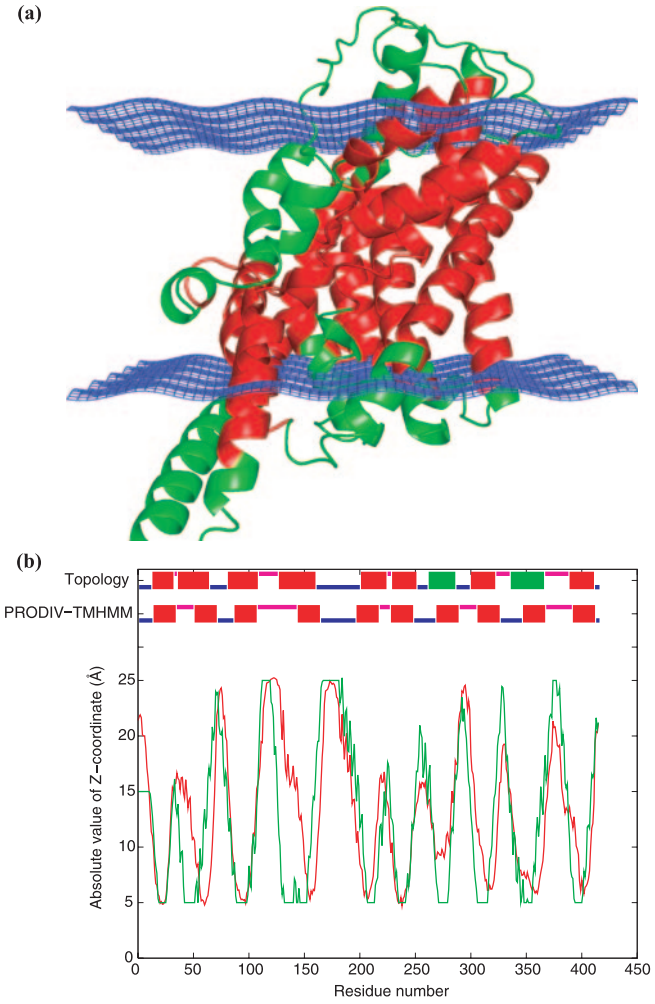


Fig. 3. Z–coordinate prediction of the glutamate transporter homolog from *Pyrococcus horikoshi*. **(a)** Residues predicted by ZPRED to be within 15Å from the center of the membrane are colored red, the rest are colored green. The blue mesh is located at ±15Å. **(b)** Correct and predicted topologies (top), where inside loops are colored blue, outside magenta, TM helices red and re-entrant regions green. The Z–coordinates are shown below with the target Z–coordinate colored green and the ZPRED prediction red.

HMM training

The HMM-based topology predictors TMHMM2.0 (Krogh *et al*., 2001), PRO-TMHMM and PRODIV-TMHMM (Viklund and Elofsson, 2004) were adjusted to emit Z–coordinates using the following procedure.

A sequence profile was constructed for each sequence by running BLAST (Altschul *et al*., 1990) with an e-value cutoff of 10^{-5}. The most probable state path for each sequence was attained using the geometric mean (GM) extension to the Viterbi algorithm. During this stage the sequences were provided with labels (M, i, o) to ensure the most probable path to be consistent with the correct topology. The labels were loosened by 10 states around each region border to allow the model some freedom in adjusting the position of the membrane regions. Each state was then assigned a

Z–coordinate by calculating the mean value of the *Z*–coordinates for the residues in the dataset that were emitted in that state.

When predicting *Z*–coordinates for a sequence, the most probable state path for that sequence is calculated using the Viterbi algorithm together with unlabeled sequences and translated into *Z*–coordinates using the estimated values for each state.

When evaluating the prediction performance, a strict jackknifing procedure was used, i.e. the state *Z*–coordinates used when evaluating the performance of a particular sequence were estimated using all sequences except the one being tested.

The six class predictions made from the ZPRED-D neural networks were encoded as a second discrete alphabet and added to the HMMs. Topology predictions are performed using the neural network outputs as a profile input vector to the HMM alongside the regular amino acid profile vector. The state emission score is calculated as the joint score of the amino acid profile and the *Z*–coordinate class profile:

$$\prod_{i=1}^{A} e(a_i)^{X(a_i)} * \prod_{j=1}^{Z} e(z_j)^{X(z_j)},$$

where the first product is the GM state score for the amino acid vector ($e(a_i)$ is the emission probability value and $X(a_i)$ is the corresponding profile vector value) and the second product is the GM state score for the *Z*–coordinate class vector ($e(z_j)$ is the emission probability value and $X(z_i)$ is the profile vector value). The state emission parameters for the *Z*–coordinate classes were optimized using simulated annealing.

Topology assignment from the *Z*–coordinate

All residues predicted below 10Å were annotated as membrane helix. A membrane region of 10Å implies that a transmembrane helix is ∼13 residues, which is substantially smaller than the 20 residues needed to traverse a 30Å thick membrane bilayer. However, having a cutoff at 15Å would miss many short loops. A filter that splitted helix regions longer than 25 residues in half and removing helix regions shorter than 4 residues was also applied. The inside and outside annotation of the loop was done by calculating the number of positive charges (Arginine and Lysine) 10 residues from the helix start or end and 5 residues into the helix, i.e. the "positive inside" rule (von Heijne, 1986, 1994). The positive charges were summed for every other loop, with the largest sum set as "inside" and the opposite side as "outside".

Predictions were evaluated on the sequence level where a topology is considered correctly predicted if all membrane regions are detected with a minimum overlap of 5 residues compared to the correct topologies and the orientation of the loop regions is correct.

ACKNOWLEDGEMENTS

This work was supported by grants from the Swedish Natural and Medical Sciences Research Council and the Wallenberg Consortinum North.

REFERENCES

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N. and Weissig,H. (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.

Bishop,C.M. (1995) *Neural networks for pattern recognition*. Oxford university press.

Chen,C.P. and Rost,B. (2002) State-of-the-art in membrane protein prediction. *Appl. Bioinformatics*, **1**, 21–35.

Chen,C.P., Kernytsky,A. and Rost,B. (2002) Transmembrane helix predictions revisited. *Protein Sci.*, **11**, 2774–2791.

Engelman,D.M., Steitz,T.A. and Goldman,A. (1986) Identifying nonpolar transbilayer helices in amino acid sequencesof membrane proteins. *Annu. Rev. Biophys. Biophys. Chem.*, **15**, 321–353.

Granseth,E., Daley,D.O., Rapp,M., Melen,K. and von Heijne,G. (2005a) Experimentally constrained topology models for 51,208 bacterial inner membrane proteins. *J. Mol. Biol.*, **352**, 489–494.

Granseth,E., von Heijne,G. and Elofsson,A. (2005b) A study of the membrane-water interface region of membrane proteins. *J. Mol. Biol.*, **346**, 377–385.

Hedman,M., Deloof,H., von Heijne,G. and Elofsson,A. (2002) Improved detection of homologous membrane proteins by inclusion of information from topology prediction. *Protein Sci.*, **11**, 652–658.

Käll,L. and Sonnhammer,E.L. (2002) Reliability of transmembrane predictions in whole-genome data. *FEBS Lett.*, **532**, 415–418.

Krogh,A., Larsson,B., von Heijne,G. and Sonnhammer,E.L. (2001) Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes. *J. Mol. Biol.*, **305**, 567–580.

Popov,M., Tam,L.Y., Li,J. and Reithmeier,R.A. (1997) Mapping the ends of transmembrane segments in a polytopic membrane protein. scanning N-glycosylation mutagenesis of extracytosolic loops in the anion exchanger, band 3. *J. Biol. Chem.*, **272**, 18325–18332.

Sonnhammer,E.L., von Heijne,G. and Krogh,A. (1998) A hidden markov model for predicting transmembrane helices in protein sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **6**, 175–182.

Taylor,W.R., Jones,D.T. and Green,N.M. (1994) A method for alpha-helical integral membrane protein fold prediction. *Proteins*, **18**, 281–294.

Tornroth-Horsefield,S., Wang,Y., Hedfalk,K., Johanson,U., Karlsson,M., Tajkhorshid,E., Neutze,R. and Kjellbom,P. (2006) Structural mechanism of plant aquaporin gating. *Nature*, **439**, 688–694.

Tusnády,G.E., Dosztanyi,Z. and Simon,I. (2005) TMDET: web server for detecting transmembrane regions of proteins by using their 3d coordinates. *Bioinformatics*, **21**, 1276–1277.

Tusnády,G.E. and Simon,I. (1998) Principles governing amino acid composition of integral membrane proteins: Application to topology prediction. *J. Mol. Biol.*, **283**, 489–506.

Tusnády,G.E. and Simon,I. (2001) The HMMTOP transmembrane topology prediction server. *Bioinformatics*, **17**, 849–850.

Viklund,H. and Elofsson,A. (2004) Best alpha-helical transmembrane protein topology predictions are achieved using hidden markov models and evolutionary information. *Protein Sci.*, **13**, 1908–1917.

von Heijne,G. (1986) The distribution of positively charged residues in bacterial inner membrane proteins correlates with the trans-membrane topology. *EMBO J.*, **5**, 3021–3027.

von Heijne,G. (1992) Membrane protein structure prediction. Hydrophobicity analysis and the positive-inside rule. *J. Mol. Biol.*, **225**, 487–494.

von Heijne,G. (1994) Membrane proteins: From sequence to structure. *Biophys. Biomol. Struct.*, **23**, 167–192.

White,S.H. (2004) The progress of membrane protein structure determination. *Protein Sci.*, **13**, 1948–1949.

White,S.H. and Wimley,W.C. (1999) Membrane protein folding and stability: physical principles. *Annu. Rev. Biophys. Biomol. Struct.*, **28**, 319–365.

Yernool,D., Boudker,O., Jin,Y. and Gouaux,E. (2004) Structure of a glutamate transporter homologue from *Pyrococcus horikoshii*. *Nature*, **431**, 811–818.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e197–e202
doi:10.1093/bioinformatics/btl257

Hairpins in a Haystack: recognizing microRNA precursors in comparative genomics data

Jana Hertel[1,*] and Peter F. Stadler[1,2,3]

[1]Bioinformatics Group, Department of Computer Science and Interdisciplinary Center for Bioinformatics, University of Leipzig, Härtelstraße 16-18, D-04107 Leipzig, Germany,
[2]Institute for Theoretical Chemistry, University of Vienna, Währingerstraße 17, A-1090 Wien, Austria and
[3]Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA

ABSTRACT

Summary: Recently, genome-wide surveys for non-coding RNAs have provided evidence for tens of thousands of previously undescribed evolutionary conserved RNAs with distinctive secondary structures. The annotation of these putative ncRNAs, however, remains a difficult problem. Here we describe an SVM-based approach that, in conjunction with a non-stringent filter for consensus secondary structures, is capable of efficiently recognizing microRNA precursors in multiple sequence alignments. The software was applied to recent genome-wide RNAz surveys of mammals, urochordates, and nematodes.

Availability: The program `RNAmicro` is available as source code and can be downloaded from http://www.bioinf.uni-leipzig/Software/RNAmicro

Contact: Jana Hertel, Tel: ++49 341 97 16704, Fax: ++49 341 97 16709, {jana,studla}@bioinf.uni-leipzig.de

1 INTRODUCTION

MicroRNAs (miRNAs) form an abundant class of non-coding RNA genes that have an important function in post-transcriptional gene regulation and in particular modulate the expression of developmentally important genes in both multi-cellular animals and plants. In both kingdoms they act as negative regulators of translation. They are transcribed as longer primary transcripts from which approximately 70 nt precursors (pre-miRNAs) with a characteristic stem-loop structure are extracted; after export to the cytoplasm, the mature miRNAs, approximately 22 nt in length, are cut out from one side of the precursor stem structure. For reviews on the discovery and function of miRNAs we refer to the literature, see e.g. (Ambros, 2004; Kidner & Martienssen, 2005). At present, several hundred distinct miRNA families are known in metazoan animals (Griffiths-Jones *et al*., 2005; Hertel *et al*., 2006), and a few dozens have been described in plants (Griffiths-Jones *et al*., 2005; Zhang *et al*., 2005; Axtell & Bartel, 2005). In contrast to other major RNA classes, in particular tRNAs, there is no recognizable homology between different families, so that it is unclear whether they arose independently in evolution or whether they derive from a single ancestral microRNA gene.

There are two basic strategies to detecting novel miRNAs. The simpler one uses sequence homology to experimentally known miRNAs as well as the characteristic hairpin structure of the pre-miRNA (Weber, 2005; Legendre *et al*., 2005; Hertel *et al*., 2006; Dezulian *et al*., 2006). A specialized machine learning approach that is specifically designed to search for distant homologs of human miRNA families is described in (Nam *et al*., 2005). Clearly, this approach is not capable of finding miRNAs for which no family member is already known.

Several approaches have focused on detecting novel miRNAs based on the secondary structure of their precursor, sequence conservation in related organisms, and the sequence conservation patterns of the 3′ and 5′ arms precursor hairpin. The programs `miRscan`[1] (Lim *et al*., 2003b), `miRseeker` (Lai *et al*., 2003), and `miralign`[2] (Wang *et al*., 2005) have lead to the discovery of a large number of novel microRNAs in nematodes (Lim *et al*., 2003b), insects (Lai *et al*., 2003;Wang *et al*., 2005) and vertebrates (Lim *et al*., 2003a). Grad *et al*., (2003) developed a computational method for predicting miRNAs in the *C. elegans* genome using both sequence and structure homology with known miRNAs. A similar procedure was employed in the plant-specific `harvester` approach (Dezulian *et al*., 2006). Berezikov *et al*. (2005) use phylogenetic shadowing to find regions that are under stabilizing selection and exhibit the characteristic variations in sequence conservation between stems, loop, and mature miRNA. In this case, secondary structure is used in a later filtering step. Genomic context also can give additional information: `Mirscan-II`, for example, takes conservation of surrounding genes into account (Ohler *et al*., 2004). Altuvia *et al*., (2005) utilize the propensity of miRNAs to appear in genomic clusters (often in the form of polycistronic transcripts) as an additional selection criterion.

MicroRNA detection without the aid of comparative sequence analysis is a very hard task but unavoidable when species-specific miRNAs are of prime interest. The `miR-abela`[3] approach first searches for hairpins that are robust against changes in the folding windows (and also thermodynamically stabilized) and then uses a support vector machine (SVM) to identify microRNAs among these candidates (Sewer *et al*., 2005). A related technique is described by Xue *et al*. (2005). The program `PalGrade` scores hairpins in a somewhat similar way (Bentwich *et al*., 2005). A quite different

*To whom correspondence should be addressed.

[1]http://genes.mit.edu/mirscan/
[2]http://bioinfo.au.tsinghua.edu.cn/miralign
[3]http://www.mirz.unibas.ch/cgi/pred_miRNA_genes.cgi

approach starts with the analysis of overrepresented patterns in phylogenetic footprints located in the 3′UTRs of mRNAs. These motifs constitute putative microRNA target sites and are used to guide the search for corresponding pre-miRNA candidates (Xie *et al*., 2005).

Advances in computational RNomics have most recently made it feasible to perform genome-wide surveys for non-coding RNAs that are not *a priori* restricted to particular RNA classes. Programs such as `qrna` (Rivas & Eddy, 2001), `EvoFold` (Pedersen *et al*., 2006), and `RNAz` (Washietl *et al*., 2005b) attempt to discover evolutionarily conserved RNA secondary structures in given multiple sequence alignments. Two distinct approaches have been realized: `EvoFold` and `qrna` are based on SCFGs (stochastic context free grammars) to evaluate the probability that the aligned sequences have evolved under the constraint of conserving secondary structure. `RNAz`, in contrast, is based on energy-directed RNA folding and assesses both thermodynamic stabilization of the secondary structure relative to a randomized control and structural conservation as measured by the relative folding energy of an alignment consensus consensus (Hofacker *et al*., 2002). A support vector machine (SVM) is then employed to classify the multiple sequence alignment as ''structured RNA''. Both `RNAz` and `Evofold` have been applied to surveying the human genome providing evidence for tens of thousands of genomic loci with signatures of evolutionarily conserved secondary structure (Washietl *et al*., 2005b; Pedersen *et al*., 2006) and detected tens of thousands of putative structured RNAs. Further `RNAz` surveys have been conducted for urochordates (Missal *et al*., 2005), nematodes (Missal *et al*., 2006), and yeasts (Steigele *et al*., 2006).

These surveys produced extensive lists of candidates for functional RNAs without using (or providing) information on membership in a particular class of RNAs. The large number of putative ncRNAs (from a few thousands in invertebrates to about 100000 in mammals) prompts the development of efficient automatic tools for their further classification and annotation.

With the exception of a small number of evolutionarily very well conserved RNAs (in particular rRNAs, tRNAs (Lowe & Eddy, 1997), the U5 snRNA (Collins *et al*., 2004), RNAse P and MRP (Piccinelli *et al*., 2005)), most ncRNAs are not only hard to discover *de novo* in large genomes, but they are also surprisingly hard to recognize if presented without annotation. Indeed, given an alignment not more than a few hundred nucleotides in length that is known to contain an conserved secondary structure, it should be very easy to decide whether these sequences belong to a known class of ncRNAs or not. Conceptually, this is a simple classification task that should be solvable efficiently by most machine learning techniques.

In the case of non-coding RNAs, however, machine learning approaches severely suffer from the very limited amount of available positive training data and the fact that negative training data are almost never known at all. Even for the most benign case, microRNA precursors, there is only a few hundred independent known examples, namely the miRNA families listed in the `mir-base` (Griffiths-Jones, 2004; Griffiths-Jones *et al*., 2005; Hertel *et al*., 2006). Over-training is thus a serious problem. As a consequence, it is necessary to restrict oneself to a small set of descriptors. This constraint, however, makes the choice of the descriptors a crucial task. Since most ncRNAs have well-conserved secondary structures, it seems natural to include

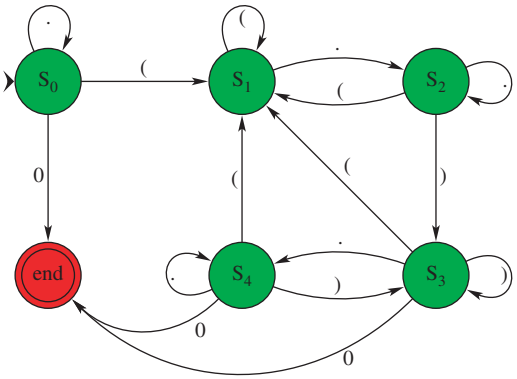


Fig. 1. Secondary structure automaton. The automaton reads an RNA secondary structure string in dot parantheses notation, recognizes all substructures, and stores their start positions and lengths.

structural descriptors in the classification procedure. RNA structure prediction, however, is less than perfect even when covariation information from the alignments can be used (Hofacker *et al*., 2002). This is true in particular when the exact ends of structured sequence within the multiple sequence alignment are not known.

In this contribution we present an SVM-based classificator for microRNA precursors that is designed to evaluate the information contained in multiple sequence alignments. The program `RNAmicro` is designed specifically to work as a 'sub-screen' for large-scale ncRNA surveys with `RNAz` or `Evofold`. The goal of `RNAmicro` is thus a bit different from that of specific surveys for miRNAs in genomic sequences: in the latter case one is interested in very high specificity so that the candidates selected for experimental verification contain as few false positives as possible. `RNAmicro`, in contrast, tries to provide an annotation of the `RNAz` survey data, so that we are interested in a more balanced trade-off between sensitivity and specificity similar to that of annotating protein motifs in known predicted protein coding genes.

2 METHODS

`RNAmicro` consists of (1) a preprocessor that identifies conserved 'almost-hairpins' in a multiple sequence alignment, (2) a module that computes a vector of numerical descriptors from each 'almost-hairpin', and (3) a support vector machine used to classify the candidate based on its vector of descriptors.

2.1 Detecting 'Almost Hairpins'

The outer loop of `RNAmicro` extracts windows of length L in 1-nucleotide steps from the input alignment. For each window, consensus sequence and consensus structure are computed using the `RNAalifold` algorithm (Hofacker *et al*., 2002) implemented in the `Vienna RNA Package` (Hofacker *et al*., 1994; Hofacker, 2003). The automaton in Fig. 1 is then used to analyze the consensus secondary structure, which is obtained in 'dot-parenthesis' notation[4].

Alignment windows whose consensus structure does not contain a stem with at least 10 base pairs or which contains two or more hairpins with at least 5 base pairs each are classified as 'not a miRNA precursor' without further analysis. Otherwise, the starting position and the length ℓ of the

[4]In this string notation for secondary structures, each unpaired nucleotide is represented by a dot, while base pairs correspond to matching pairs of parentheses.

Table 1. Descriptors used for SVM classification

Property	#	Descriptors
Structure	2	l_s, l_h
Sequence composition	1	G+C
Sequence conservation	4	$S_{5'}$, $S_{3'}$, S_0, $S_{\min}$
Thermodynamic stability	4	$\bar{E}$, $\bar{\epsilon}$, $\bar{\eta}$, $\bar{z}$
Structure conservation	1	E_{cons}
Total	12	

See text for definitions.

'almost-hairpin' which constituted the pre-miRNA candidate, are recorded and the corresponding alignment window is used to compute the descriptors. This filter, which on purpose is not very stringent, thus also accepts stem-loop structures with short 'branches' as candidates. Some important animal microRNAs are known to have structures of this type, for example *let-7*.

2.2 Descriptors

The lengths l_s and l_h of stem and hairpin loop regions recognized by the automaton form the first two descriptors provided the alignment window passes the structure filter. In addition we use the G+C content.

The second class of descriptors summarizes the thermodynamic properties of local sequence interval. MicroRNA precursors are known to be more stable than other RNAs with the same sequence composition (Bonnet *et al*., 2004; Clote *et al*., 2005). We thus use the average $\bar{z}$ of the energy z-scores

$$z = (E - \langle E \rangle_{\text{random}})/\sigma \tag{1}$$

where E is the folding energy of the given sequence. The mean $\langle E \rangle_{\text{random}}$ and σ of the distribution of randomized sequences is computed from a regression model as described by Washietl *et al*. (2005b) instead of using a shuffling procedure. Zhang *et al*. (2006) reported two folding energy scores that efficiently distinguish pre-miRNAs from other ncRNAs. The '*adjusted mfe*' is defined as $\epsilon = 100 \times E/\ell$; the '*mfe index*' η is the ratio of ϵ and the G+C content. We use their average values $\bar{\epsilon}$ and $\bar{\eta}$ as descriptors.

Structural conservation can be assessed by the *structure conservation index* (Washietl *et al*., 2005b), i.e. the ratio of the average folding energy of the aligned sequences and the energy of the consensus secondary structure. We use here $\bar{E}$ and E_{cons} separately.

An important characteristic of pre-miRNAs is the difference in the sequence conservation between the mature miRNA, which may be contained at either the 3′ or the 5′ side of the stem-loop structure, other parts of the stem, and the hairpin loop region, respectively, see e.g. (Lim *et al*., 2003b; Lai *et al*., 2003). We compute the average columnwise entropies $S_{5'}$, $S_{3'}$, and S_0, separately for 5′ and 3′ sides of the stem region and the hairpin loop. For a region (i.e., a subset of alignment positions) ξ we define

$$S_\xi = -\frac{1}{\text{len}(\xi)} \sum_{i \in \xi} \sum_{\alpha = A, C, G, U} p_{i,\alpha} \ln p_{i,\alpha} \tag{2}$$

where $p_{i,\alpha}$ is the fraction of α nucleotides at sequence position i. Since the mature miRNA is typically extremely well conserved, we determine the sequence window of length 23 with the lowest entropy $S_{\min}$ and use this value as an additional descriptor, Table 1.

2.3 SVM implementation

For classification we used a support vector machine as implemented in the `libsvm` package, version 2.8, (Chang & Lin, 2001). Descriptor vectors were scaled linearly to the interval $[-1, +1]$ before training using the binary version of `svm-scale` which is included in the `libsvm` package. The SVM was then trained using a radial basis function (RBF) kernel with $\gamma = 2$ and probability estimates. Default settings as listed in the README file of the `libsvm` package were used for all other parameters. The RBF kernel was used based on the recommendation of the `libsvm` documentation and positive experience with this kernel in the `RNAz` program. As we shall see below, these settings give satisfactory results in our context.

For alignments of length at most L, a single classification is performed. For longer alignments, we used a sliding window of length L with step-size 1. In this case, only the best (w.r.t. to SVM classification confidence value p) non-overlapping windows of length L were retained for each input alignment.

Table 2. Initial training and performance of RNAmicro SVM

Classification	Test sets *Positive*	*Negative*
miRNA	134	2
not miRNA	13	381
Total	147	383

Half of the positive and negative sets were used for training and testing, respectively.

2.4 SVM Training

Due to the relative sparseness of the available training data we used a stepwise training scheme. The positive training set is constructed from the union of animal microRNAs contained in the `miRNA registry` 6.0 and orthologous and paralogous sequences that have been obtained by a homology search in all metazoan genomes (Hertel *et al*., 2006). This set consisted of 295 alignments of distinct microRNA families composed by 2 up to 20 sequences from nematodes, insects, and vertebrates. Care was taken to avoid any sequence similarity between different alignments by using the family definition of (Hertel *et al*., 2006), which identifies several groups of microRNAs with different `mirbase` numbers as homologs. The antagonistic data was obtained by randomly shuffling the columns of each *true* miRNA alignment until the consensus sequence of the shuffled alignment folded again into a hairpin structure. This was successful for all but one true miRNA alignment. We have to rely at least in part on artificial examples since it seems hard to obtain a large collection of mutually independent evolutionarily conserved hairpin structures that are *known* not to be pre-miRNAs. The artificial set of negatives was complemented by a collection of 483 tRNA alignments which also passed the hairpin check. Note, however, that tRNAs are fairly similar to each other and hence cover only a relatively small part of the descriptor space.

In order to assess the quality of the descriptors, we divided both the positive and the negative set randomly into two halves, one used for training the SVM and the other used as test set. Consequently, there was no significant phylogenetic bias in the training set versus the test set.

We used `RNAmicro` with three different window sizes, $L = 70, 100, 130$, to scan the input alignments. An alignment is classified as putative microRNA if at least one window of at least one of the three values of L is classified with $p > 0.5$ by the SVM. We achieve a sensitivity of about 90% (134/147) and a specificity of about 99% (381/383) on the test dataset, Table 2. As an alternative training and testing we divided the available data into 90% for training and tested if the remaining 10% were classified correctly. This yields in a sensitivity of about 84% (26/31) and a specificity of about 99% (153/155).

Since the different training schemes yield consistent results and the training and test alignments are unrelated at sequence level, over-training thus does not seem to be a serious issue. We therefore trained the SVM using the entire positive and negative sets. We then tested the program on the results of `RNAz` screens of nematodes (Missal *et al*., 2006) and seasquirts (Missal *et al*., 2005). Although we could classify almost all known miRNAs that were contained in these data as miRNA, we found that in addition a

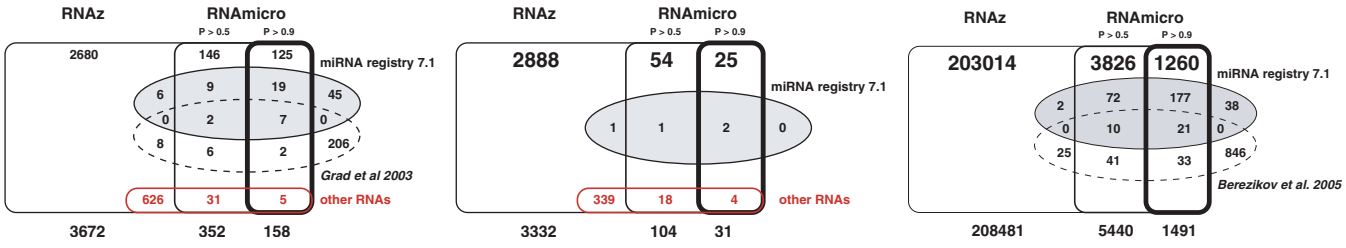


Fig. 2. Venn diagrams of RNAmicro-classifications of RNAz survey data with a RNAz cutoff of 0.5. The subsets of structured RNAs that are classified as miRNA candidates by RNAmicro are shown with bold outlines for $p = 0.5$ and $p = 0.9$ confidence levels. The subset of known microRNAs are shown with a grey background. Red numbers are other known ncRNAs or UTR elements that constitute known false positives in the $0.5 < p \leq 0.9$ and the $p > 0.9$ confidence classes, respectively. Numbers below the Venn diagram are the total number of RNAz alignments that were screened by RNAmicro, and the total numbers of signals classified as positive at confidence values $p = 0.5$ and $p = 0.9$, respectively. (a) Data from a pairwise screen of the nematoda *C. elegans* and *C. briggsae* (Missal *et al.,* 2006). In this case many known ncRNAs are contained in the data set allowing at least a rough estimate of false positive rates. (b) In the case of the two urochordates *Ciona intestinalis* and *Ciona savignyi* only 4 miRNAs are known. (c) For the screen of mammalian genomes comprising sequences that are conserved at least in human, dog, mouse, and rat (Washietl *et al.,* 2005a) almost all known non-coding RNAs were not available in the input alignments because they are marked as repetitive (tRNAs, snRNA, some microRNAs), so that a meaningful estimate for the false positive rate cannot be derived.

significant number of other known ncRNAs was mis-classified as pre-miRNAs. This indicates that our initial negative set does not sufficiently cover the descriptor space. The reason is that hairpins are common motifs in many other ncRNAs and that several other ncRNA families are also known to be thermodynamically very stable (Clote *et al*., 2005).

We therefore extracted alignments of noncoding RNAs from the Rfam database, focusing on a subset of snoRNAs, rRNAs, additional tRNAs, and RNAseP sequences and scored those with RNAmicro. False positives were added to the negative set and RNAmicro was retrained and tested with the 50% method as described above. The sensitivity was still around 90% while the specificity dropped to 78%. Thus, the mis-classified alignment slices of the negative input alignments were added to the training set. This procedure was iterated until no significant improvement was achieved on the Rfam dataset. This procedure is not statistically sound, of course. The alignments from the RNAz surveys contain in part different combinations of species and have been produced with different methods than those used for training, so that we can at least check the sensitivity of the model on the RNAz-alignments of the known microRNA precursors. Furthermore, other known ncRNAs in these data serve as a negative control.

3 APPLICATIONS

Three extensive surveys of metazoan genomes using RNAz (Washietl *et al*., 2005b) have been published recently. The screen of vertebrate genomes (Washietl *et al*., 2005a) was based on the top 5% conserved multiz alignments (Blanchette *et al*., 2004) as determined by phastcons (Siepel *et al*., 2005). For nematodes and urochordates, alignments were constructed using clustalw based on initial blast hits, see (Missal *et al*., 2005, 2006) for details. In all three cases, only non-repetitive non-protein-coding sequences were investigated.

In order to identify putative miRNAs among them we screened all individual alignment slices that were classified as potentially structured RNA with SVM classification confidence of $p_{\text{RNAz}} > 0.5$. Note that in all three studies individual alignment slices are combined to single 'RNAz hits' when they overlapped on the genome of the species. Hence the number of alignment slices is much larger than the number of 'RNAz hits' reported in these studies. Redundancies arising from miRNAs that appear in more than one alignment slice have been removed. The Venn diagrams in Fig. 2 summarize our classification.

It is reassuring that most of the RNAmicro predictions have high confidence values in the original RNAz screens: For example, 3850 (70%) of the 5440 $p_{\text{RNAmicro}} > 0.5$ candidates in the mammalian screen have $p_{\text{RNAz}} > 0.9$. Conversely, Only 204 (14%) of the 1491 $p_{\text{RNAmicro}} > 0.9$ have $p_{\text{RNAz}} < 0.9$. At least a rough estimate for the false discovery rate can be obtained from the distribution of the classification confidence values. For the three RNAz surveys we expect that about 1/5 to 1/4 of the putative ncRNAs are false positives at $p > 0.5$ classification confidence (not shown).

Berezikov *et al*. (2005) predicted 976 miRNAs by scanning whole-genome human/mouse and human/rat alignments. Their method, however, highlights evolutionary recent microRNAs so that it is not too surprising that there is relatively little overlap between these candidates and the RNAz screen (Washietl *et al*., 2005a), which focuses on evolutionary well-conserved RNA structures.

In order to compare our prediction with related classification methods, we re-evaluated the positive RNAmicro predictions using the SVM approach by Xue *et al*. (2005), which is designed for finding miRNAs *ab initio* in genomic sequences. Their procedure employs a very restrictive check for hairpin structures which in particular rejects the majority (180) of the 249 known microRNA precursors. Only 3077 of our 5440 $p > 0.5$ candidates and only 953 of our 1481 $p > 0.9$ candidates pass the hairpin filter. Of these, 1590 and 657, resp., are scored as microRNAs. Screening the $p_{\text{RNAz}} \geq 0.9$ subset with mir-abela returned 981 candidates, of which RNAmicro classifies 515 as microRNA precursors.

Several computational searches for miRNAs have been performed for nematodes. Grad *et al*. (2003) predicted 222 microRNA candidates (beyond those known at the time of publication) for *C. elegans*. Since most of the candidates are not conserved in *C. briggsae*, these sequence were not in the input set of RNAz survey. Thus, this set shows little overlap with our classification. Nevertheless it is interesting to note that the estimated total number of miRNAs is comparable. In contrast, based on the

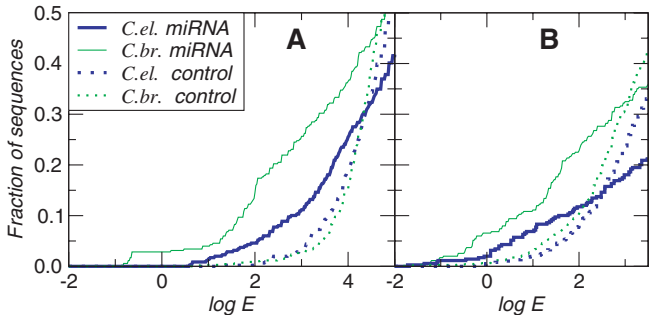


Fig. 3. Distribution of two closely related upstream motifs (A) and (B) reported for both *C. elegans* and *C. briggsae* (Ohler *et al.*, 2004, Fig.2). We plot the fraction of RNAmicro candidates for which mast (Bailey & Gribskov, 1998) recovers at least one copy A or B within 2000 nt upstream of the miRNA candidate as a function of the mast *E*-value cutoff. For small cutoffs, the miRNA specific sequence elements are overrepresented in true data versus a control set of RNAz hits that were not classified as microRNAs.

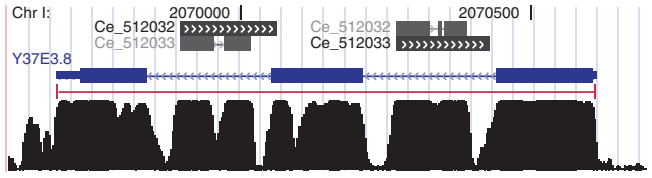


Fig. 4. Typical example of a pair of related putative intronic microRNAs in ***C. elegans*** extracted from the USCS genome browser. The gene ***Y37E3.8*** is a hypothetical protein of unknown function. The 'mountain range' on the bottom displays the sequence conservation between *C. elegans* and *C. briggsae*.

results of experimental verification of mirscan predictions, Lim *et al*. (2003b) and Ohler *et al*. (2004) conclude that the overwhelming majority of *C. elegans* miRNAs should have been found already.

Ohler *et al*. (2004) reported upstream sequence motifs specific to independently transcribed miRNAs in *C. elegans* and *C. briggsae*. We have therefore searched 2000 nt upstream for approximate occurrences of these patterns using mast. We find that both approximate patterns are substantially overrepresented in sequences classified as miRNAs relative to the remainder of the data, (Fig. 3). This provides additional *statistical* evidence that a substantial fraction of the RNAmicro-predictions indeed are microRNAs. As noted by Ohler *et al*. (2004), these sequence patterns, which are presumably transcription factor binding sites, do not occur associated with intronic miRNAs. We find that 176 (50%) of the 351 *C. elegans* candidates are located in introns (Fig. 4).

In the human data, 4245 candidates are not associated with known protein-coding genes, while 1107 candidates (20%) are located in introns (of which 36 are known microRNAs). This is in agreement with a recent study reporting that intronic microRNAs are much more frequent than previously thought (Ying & Lin, 2006). The remaining 88 sequences map to exons of known genes and are probably false positives.

MicroRNAs have a tendency to appear in clusters, probably because they are frequently processed from a polycistronic transcript. This fact has been utilized by (Altuvia *et al*., 2005; Sewer *et al*., 2005) to identify additional miRNAs in the vicinity of known ones. Using a rather conservative distance cutoff of <1000 nt between adjacent miRNAs, we found 143 clusters of miRNA candidates in the human genome, which contain 316 individual candidate sequences. Among them are 58 known miRNAs (according to mirbase 7.1) in 33 clusters. Most prominently, we recover the extensive imprinted cluster at human locus 14q32 discovered by (Lagos-Quintanta *et al*., 2002) (in total, we found 54 candidates in multiple tight clusters between positions 100M and 101M of the hg17 assembly) and the paralogs of the *mir-17* cluster (Tanzer & Stadler, 2004). In *C. elegans* we find 30 clusters with 131 members, in *C. intestinalis* there are 5 clusters with 10 members. Note that these are conservative estimates since in some cases, such as the *C. elegans mir-42* cluster, it is known that the distance between clustered miRNAs can be larger.

4 DISCUSSION

In contrast to other related approaches to miRNA detection, RNAmicro does not directly search a genome or genomes. Instead it is designed to classify the raw results of large-scale comparative genomics surveys for putative RNAs that are *conserved in both sequence and secondary structure*. Consequently, RNAmicro uses a different tradeoff between sensitivity and specificity. In the spirit of protein annotation methods, we aim for very high sensitivity rather than minimizing the expected number of false positives. As classificators become available for other classes of ncRNAs and common UTR motifs, conflicting class assignments from different classificators will eventually help to improve the specificity of miRNA detection.

Clearly, the performance of RNAmicro depends on the sensitivity and specificity of the initial screen for structured RNA candidates. However, RNAz exhibits a sensitivity of more than 80% at 99% specificity already on pairwise alignments (Washietl *et al*., 2005b, Table 2). In practice, it recovered 157 of the 163 human microRNAs in the input alignments that were known when the RNAz survey was performed (Washietl *et al*., 2005a). We therefore argue that this first step does not dramatically influence the overall sensitivity for microRNAs. Instead, the main limitations rather lie in (a) the coverage and quality of the input alignments and (b) the phylogenetic conservation of microRNAs, which of course limits all comparative approaches.

We have applied RNAmicro to three recent RNAz-bases studies of mammalian, nematode, and urochordate ncRNAs. In each case a large number of novel miRNA candidates have been detected. We have therefore investigated whether there is confounding evidence that a significant fraction of these predictions should be true positives: In *C. elegans*, for example, we find a strong association of RNAmicro predictions with a miRNA specific upstream motif previously reported by Ohler *et al*. (2004).. Furthermore, we found several hundred miRNA candidates that occur in tight genomic clusters. In particular in the human data, a large number of predictions are located within 1000 nt of a known microRNA. In line with recent reports (Ying & Lin, 2006),, we furthermore observed a substantial fraction (20% in human, 50% in *C. elegans*) of candidates are located in introns. Thus we argue that a large part of the RNAmicro candidates corresponds to real microRNAs. It is well conceivable that we have seen only a small fraction of the true miRNA repertoire to due to small expression levels and expression

patterns restricted to a few cell-lines (Ambros, 2004; Bartel & Chen, 2004; Mattick, 2004).

ACKNOWLEDGEMENTS

Financial support by the German *DFG* in the framework of the Bioinformatics Initiative (BIZ-6/1-2) and the SPP 'Metazoan Deep Phylogeny' is gratefully acknowledged.

REFERENCES

Altuvia,Y., Landgraf,P., Lithwick,G., Elefant,N., Pfeffer,S., Aravin,A., Brownstein,M.J., Tuschl,T. and Margalith,H. (2005) Clustering and conservation patterns of human microRNAs. *Nucleic Acids Res.*, **33**, 2697–2706.

Ambros,V. (2004) The functions of animal microRNAs. *Nature*, **431**, 350–355.

Axtell,M.J. and Bartel,D.P. (2005) Antiquity of microRNAs and their targets in land plants. *Plant Cell*, **17**, 1658–1673.

Bailey,T.L. and Gribskov,M. (1998) Combining evidence using *p*-values: application to sequence homology searches. *Bioinformatics*, **14**, 48–54.

Bartel,D.P. and Chen,C.-Z. (2004) Micromanagers of gene expression: the potentially wide-spread influence of metazoan microRNAs. *Nat. Genet.*, **5**, 396–400.

Bentwich,I., Avniel,A.A., Karov,Y., Aharonov,R., Gilad,S., Barad,O., Barzilai,A., Einat,P., Einav,U., Meiri,E., Sharon,E., Spector,Y. and Bentwich,Z. (2005) Identification of hundreds of conserved and nonconserved human microRNAs. *Nat. Genet.*, **37**, 766–770.

Berezikov,E., Guryev,V., van de Belt,J., Wienholds,E. and Ronald Plasterk,H.A. (2005) Phylogenetic shadowing and computational identification of human microRNA genes. *Cell*, **120**, 21–24.

Blanchette,M., Kent,W.J., Riemer,C., Elnitski,L., Smit,A.F.A., Roskin,K.M., Baertsch,R., Rosenbloom,K., Clawson,H., Green,E.D., Haussler,D. and Miller,W. (2004) Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.*, **14**, 708–715.

Bonnet,E., Wuyts,J., Rouzé,P. and van de Peer,Y. (2004) Evidence that microRNA precursors, unlike other non-coding RNAs, have lower folding free energies than random sequences. *Bioinformatics*, **20**, 2911–2917.

Chang,C.-C. and Lin,C.-J. (2001) LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Clote,P., Ferré,F., Kranakis,E. and Krizanc,D. (2005) Structural RNA has lower folding energy than random RNA of the same dinucleotide frequency. *RNA*, **11**, 578–591.

Collins,L.J., Macke,T.J. and Penny,D. (2004) Searching for ncRNAs in eukaryotic genomes: Maximizing biological input with RNAmotif. *J. Integ. Bioinf.*, **6**, 15p.

Dezulian,T., Remmert,M., Palatnik,J.F., Weigel,D. and Huson,D.H. (2006) Identification of plant microRNA homologs. *Bioinformatics*, **22**, 359–360.

Grad,Y., Aach,J., Hayes,G.D., Reinhart,B.J., Church,G.M., Ruvkun,G. and Kim,J. (2003) Computational and experimental identification of *C. elegans* microRNAs. *Mol. Cell*, **11**, 1253–1263.

Griffiths-Jones,S. (2004) The microRNA Registry. *Nucleic Acids Res.*, **32**, D109–D111.

Griffiths-Jones,S., Moxon,S., Marshall,M., Khanna,A., Eddy,S.R. and Bateman,A. (2005) Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res.*, **33**, D121–D124.

Hertel,J., Lindemeyer,M., Missal,K., Fried,C., Tanzer,A., Flamm,C., Hofacker,I.L. and Stadler,P.F. & The Students of Bioinformatics Computer Labs 2004 and 2005 (2006). The expansion of the metazoan microRNA repertoire. *BMC Genomics*, **7**, 25.

Hofacker,I.L., Fekete,M. and Stadler,P.F. (2002) Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, **319**, 1059–1066.

Hofacker,I.L., Fontana,W., Stadler,P.F., Bonhoeffer,L.S., Tacker,M. and Schuster,P. (1994) Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, **125**, 167–188.

Hofacker,I.L. (2003) Vienna RNA secondary structure server. *Nucleic Acids Res.*, **31**, 3429–3431.

Kidner,C.A. and Martienssen,R.A. (2005) The developmental role of microRNA in plants. *Curr. Opin. Plant Biol.*, **8**, 38–44.

Lagos-Quintanta,M., Rauhut,R., Yalcin,A., Meyer,J., Lendeckel,W. and Tuschl,T. (2002) Identification of tissue specific microRNAs from mouse. *Curr. Biol.*, **12**, 735–739.

Lai,E.C., Tomancak,P., Williams,R.W. and Rubin,G.M. (2003) Computational identification of drosophila microRNA genes. *Genome Biol.*, **4**, R42, [Epub].

Legendre,M., Lambert,A. and Gautheret,D. (2005) Profile-based detection of microRNA precursors in animal genomes. *Bioinformatics*, **21**, 841–845.

Lim,L.P., Glasner,M.E., Yekta,S., Burge,C.B. and Bartel,D.P. (2003a) Vertebrate microRNA genes. *Science*, **299**, 1540–1540.

Lim,L.P., Lau,N.C., Weinstein,E.G., Abdelhakim,A., Yekta,S., Rhoades,M.W., Burge,C.B. and Bartel,P.B. (2003b) The microRNAs of *Caenorhabditis elegans*. *Genes & Development*, **17**, 991–1008.

Lowe,T.M. and Eddy,S. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.*, **25**, 955–964.

Mattick,J.S. (2004) RNA regulation: a new genetics? *Nat. Genet.*, **5**, 316–323.

Missal,K., Rose,D. and Stadler,P.F. (2005) Non-coding RNAs in *Ciona intestinalis*. *Bioinformatics*, **21**, i77–i78.

Missal,K., Zhu,X., Rose,D., Deng,W., Skogerbø,G., Chen,R. and Stadler,P.F. (2006) Prediction of structured non-coding RNAs in the genome of the nematode *Caenorhabitis elegans*. *J. Exp. Zool.: Mol. Dev. Evol.*, DOI: 10.1002/jez.b.21086.

Nam,J.-W., Shin,K.-R., Han,J., Lee,Y., Kim,V.N. and Zhang,B.-T. (2005) Human microRNA prediction through a probabilistic co-learning model of sequence and structure. *Nucleic Acids Res.*, **33**, 3570–3581.

Ohler,U., Yekta,S., Lim,L.P., Bartel,D.P. and Burge,C.B. (2004) Patterns of flanking sequence conservation and a characteristic upstream motif for microRNA gene identification. *RNA*, **10**, 1309–1322.

Pedersen,J.S., Bejerano,G., Siepel,A., Rosenbloom,K., Lindblad-Toh,K., Lander,E.S., Kent,J., Miller,W. and Haussler,D. (2006) Classification of conserved RNA secondary structures in the human genome. *PLoS Comput. Biol.*, **2**, e33.

Piccinelli,P., Rosenblad,M.A. and Samuelsson,T. (2005) Identification and analysis fo ribonuclease P and MRP RNA in a broad range of eukaryotes. *Nucleic Acids Res.*, **33**, 4485–4495.

Rivas,E. and Eddy,S.R. (2001) Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, **2**, 8.

Sewer,A., Paul,N., Landgraf,P., Aravin,A., Pfeffer,S., Brownstein,M.J., Tuschl,T., van Nimwegen,E. and Zavolan,M. (2005) Identification of clustered microRNAs using an *ab initio* prediction method. *BMC Bioinformatics*, **6**, 267, [epub].

Siepel,A., Bejerano,G., Pedersen,J.S., Hinrichs,A., Hou,M., Rosenbloom,K., Clawson,H., Spieth,J., Hillier,L.W., Richards,S., Weinstock,G.M., Wilson,R.K., Gibbs,R.A., Kent,W.J., Miller,W. and Haussler,D. (2005) Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res*, **15**, 1034–1050.

Steigele,S., Stadler,P.F. and Nieselt,K. (2006) Computational prediction and annotation of structured RNAs in yeasts. RECOMB poster.

Tanzer,A. and Stadler,P.F. (2004) Molecular evolution of a microRNA cluster. *J. Mol. Biol.*, **339**, 327–335.

Wang,X., Zhang,J., Li,F., Gu,J., He,T., Zhang,X. and Li,Y. (2005) MicroRNA identification based on sequence and structure alignment. *Bioinformatics*, **21**, 3610–3614.

Washietl,S., Hofacker,I.L., Lukasser,M., Hüttenhofer,A. and Stadler,P.F. (2005a) Mapping of conserved RNA secondary structures predicts thousands of functional non-coding RNAs in the human genome. *Nat. Biotechnol.*, **23**, 1383–1390.

Washietl,S., Hofacker,I.L. and Stadler,P.F. (2005b) Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. USA*, **102**, 2454–2459.

Weber,M.J. (2005) New human and mouse microRNA genes found by homology search. *FEBS J.*, **272**, 59–73.

Xie,X., Lu,J., Kulbokas,E.J., Golub,T.R., Mootha,V., Lindblad-Toh,K., Lander,E.S. and Kellis,M. (2005) Systematic discovery of regulatory motifs in human promoters and 3′ UTRs by comparison of several mammals. *Nature*, **434**, 338–345.

Xue,C., Li,F., He,T., Liu,G., Li,Y. and Zhang,X. (2005) Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, **6**, 310, [epub].

Ying,S.-Y. and Lin,S.L. (2006) Current perspectives in intronic microRNAs (miRNAs). *J. Biomed. Sci.*, **13**, 5–15.

Zhang,B.H., Pan,X.P., Cox,S.B., Cobb,G.P. and Anderson,T.A. (2006) Evidence that miRNAs are different from other RNAs. *Cell. Mol. Life Sci.*, **63**, 246–254.

Zhang,B.H., Pan,X.P., Wang,Q.L., Cobb,G.P. and Anderson,T.A. (2005) Identification and characterization of new plant microRNAs using EST analysis. *Cell Res.*, **15**, 336–360.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e203–e210
doi:10.1093/bioinformatics/btl248

Modelling sequential protein folding under kinetic control

Fabien P.E. Huard[1,*], Charlotte M. Deane[2] and Graham R. Wood[1]

[1]Department of Statistics, Macquarie University, NSW 2109, Australia and [2]Department of Statistics, 1 South Park Road, Oxford OX1 3TG, UK

ABSTRACT

Motivation: This study presents a novel investigation of the effect of kinetic control on cotranslational protein folding. We demonstrate the effect using simple HP lattice models and show that the cotranslational folding of proteins under kinetic control has a significant impact on the final conformation. Differences arise if nature is not capable of pushing a partially folded protein back over a large energy barrier. For this reason we argue that such constraints should be incorporated into structure prediction techniques. We introduce a finite surmountable energy barrier which allows partially formed chains to partly unfold, and permits us to enumerate exhaustively all energy pathways.

Results: We compare the ground states obtained sequentially with the global ground states of designing sequences (those with a unique global ground state). We find that the sequential ground states become less numerous and more compact as the surmountable energy barrier increases. We also introduce a probabilistic model to describe the distribution of final folds and allow partial settling to the Boltzmann distribution of states at each stage. As a result, conformations with the highest probability of final occurrence are not necessarily the ones of lowest energy.

Availability: Software available on request

Contact: fhuard@efs.mq.edu.au

1 INTRODUCTION

There have been several definitions of cotranslational folding, but it has been elegantly stated that ''co-translational folding has occurred if, following extrusion from the ribosome, the native structure is achieved more quickly than if the full-length, unfolded polypeptide were diluted from chemical denaturant into the same folding milieu as that in which protein biosynthesis occurred'' (Baldwin, 1999). It is recognised that some proteins can fold rapidly and cotranslationally both in eukaryotic and prokaryotic cells (Basharov, 2003; Braakman *et al*., 1991; Fedorov and Baldwin, 1997; Fedorov and Baldwin, 1997; Kolb, 2001; Kolb *et al*., 2000; Netzer and Hartl, 1997) and there is recent evidence that some proteins become *in vivo* biologically active as the polypeptide chain is being translated (Nicola *et al*., 1999). We also know that cotranslational folding can occur spontaneously without additional cellular components (Sanchez *et al*., 2004). Interestingly, nitinol wire, known to remember its annealed shape, has been used to model behaviour of biopolymers and showed that in some cases the native state could only be reached sequentially (Keller, 2003).

Levinthal pointed out that the protein folding process cannot search the entire conformation space due to its vast size. Since proteins are known to fold in the order of milliseconds, we must assume that they follow a restricted set of pathways to reach their native conformation (Levinthal, 1968; Levinthal, 1969). Hence folding is assumed to be under kinetic control, that is, the folding pathway of a protein is unlikely to incorporate folding to a state which would be less thermodynamically stable. It was advanced that protein folding obeys thermodynamical laws and therefore has a native state which is the ground state of lowest free energy (Anfinsen, 1973). It has been theoretically demonstrated (Govindarajan and Goldstein, 1998) that a sequence whose native state has originally a higher energy than the lowest energy state, when submitted to evolution under kinetic control, will most often evolve towards a sequence whose native state is the lowest energy conformation. Thus folding under kinetic control does not necessarily violate the thermodynamical hypothesis.

Surprisingly, state-of-the-art protein folding prediction methods do not incorporate a cotranslational aspect (Bujnicki, 2006); in the latest Critical Assessment of Techniques for Protein Structure Prediction meeting (CASP, 2004) none of the chosen methods exploited the sequential nature of folding. Cotranslation has already been investigated in simulations of biopolymers (Bornberg-Bauer, 1997; Fernandez, 1994; Morrissey *et al*., 2004), but the effect of kinetic control remains unexplored. The method we propose aims at filling this gap; we investigate the effect of energy barriers on cotranslation.

We fold proteins sequentially, mimicking nature as closely as possible. By a ''sequential folding'' we will refer to the path of intermediate and final conformations simulated as the nascent polypeptide chain is elongated. A ''sequential ground state'' is a conformation of lowest energy obtained once all residues are added. We simulate protein fold evolution, as the polypeptide chain length increases, by sequentially elongating the length of protein to be folded, starting from the N-terminus. Amino acids are added one by one at the C-terminus of the chain and each time the chain length increases by one residue, the conformation already simulated is permitted to change. The point here is that the new fold must be a ''restricted evolution'' of the previously predicted fold. By this we mean that the simulation of the newly elongated chain does not start with a random or fully extended conformation, but with the previous model obtained as a base, to which is added the new residue. The latter is added in a fully extended conformation. We also investigate the possibility of adding more than one residue at a time. The final fold of the protein is obtained once all residues are added.

Essential here is the concept of a surmountable energy barrier (Baker, 1998; Guo *et al*., 1997; Sohl *et al*., 1998), the orchestrator of kinetic control. The surmountable energy barrier enables us to partly avoid kinetic traps, and represents the maximum energy gain

*To whom correspondence should be addressed.

possible for the protein at each step of its folding process. It is essentially the unfolding energy available in the system. In the following cases of folding under kinetic control, this surmountable energy barrier is assumed to be finite. Rationale for the imposition of a surmountable energy barrier comes from a number of sources. We know that ~20% of proteins require intervention of chaperones, which play an important role in cotranslation (Frydman, 2001; Hartl and Hayer-Hartl, 2002). It is believed that the primary role of chaperones is to prevent aggregation of nascent polypeptides. The surmountable energy barrier aims at representing the restriction on the folding pathways induced by chaperones. We also know that folding space is restricted by the structure of the ribosome itself (Ban *et al*., 2000; Ramakrishnan, 2002; Wilson *et al*., 2002). In particular, the fold of polypeptides is constrained by the ribosome exit tunnel (Jenni and Bany, 2003; Nakatogawa and Ito, 2002) which favours α-helical secondary structures (Ziv *et al*., 2005).

We know that some codons are less frequent than others, inducing different translation rates (Andersson and Kurland, 1990; Curran and Yarus, 1989) and that codon substitutions can lead to lower specific activity (Komar *et al*., 1999). Slow codons, usually positioned between domains, can induce a delay required for correct folding of the N-terminus domain (Komar and Jaenicke, 1995). Slow codons can also enhance the formation of secondary structures by preventing domains from interacting with each other (Purvis *et al*., 1987). To model the variation in translation rate imposed by codon selection, we introduce parameter s which represents the number of residues added each time the polypeptide chain is elongated. This creates a primitive ''elongate-pause'' iterative extension process.

We also attach a probability to all partial and fully extended conformations. It has been observed that the biologically active state of some proteins does not correspond to their lowest energy conformation (Sohl *et al*., 1998). We introduce a probabilistic model which captures two factors. The first factor is the number of kinetically controlled energy pathways which can lead to the conformation (relative to the number of possible conformations for the considered sequence). The second factor is the Boltzmann equilibrium distribution for the current set of partial configurations. We balance the two factors using a ''thermodynamic permission factor'' β. This measures the extent to which the Boltzmann distribution is reached. We investigate whether kinetic control together with partial movement to the Boltzmann distribution can result in a sequential ground state whose energy may be a local minimum in the thermodynamic energy path of the protein, as observed experimentally.

HP lattice models have proven a useful tool for modelling protein folding in a simple manner (Chan and Dill, 1993; Chan and Dill, 1994; Dill *et al*., 1995; Pande *et al*., 1997; Shakhnovich, 1998), predicated on the assumption that protein folding is ruled by hydrophobic collapse. Here we use them to assess the impact of sequential folding. Sequences involving only two types of monomer (hydrophobic H and polar P) are considered, with monomer positions restricted to either a two or three-dimensional lattice. Simple models have been used to simulate globular protein folding incorporating cotranslation and restrictions on the folding space, modelling the ribosome as an inert wall (Sikorski and Skolnick, 1990). It was found that α-helical proteins preferred to assemble parallel to the wall, and four member β-barrels slightly preferred assembly perpendicular to the wall. Sikorski and Skolnick ''never observed a successful case of co-translational folding'' and did not consider kinetic control. They used a Monte Carlo algorithm to search the conformation space and pass through local minima, whereas we develop a fully deterministic approach and exhaustively search the conformation space.

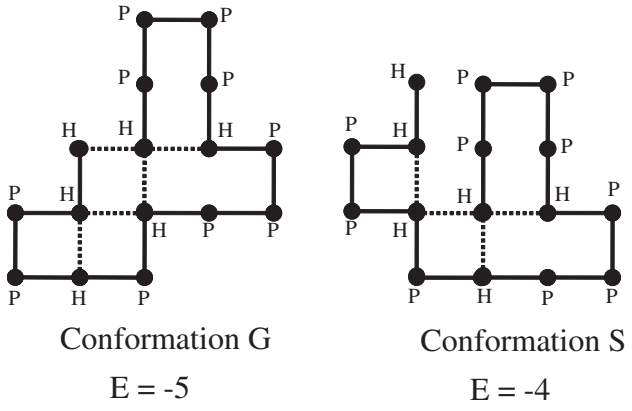


Fig. 1. Conformations obtained for the sequence HPPPPHPPPHPHPPHH. Conformation G represents the global ground state, the unique conformation which has an energy of minus five for this particular sequence. Conformation S is that obtained sequentially, with energy of minus four, when the surmountable energy barrier is zero.

In summary, we explore the consequences of following a sequential route to the final fold. In particular, we study the influence on final conformations of the height of the surmountable energy barrier d and the number of residues s added at each iteration. We find that under kinetic control the sequential ground state of a protein can differ from the global one (Figure 1). The global state of minimum energy can be reached only with a sufficiently high surmountable energy barrier.

We then present the impact of the variation of the main parameters (extrusion length and surmountable energy barrier) on the compactness and multiplicity of the folds. For a given sequence, we observe that final conformations are more compact and less numerous as we increase the surmountable energy barrier.

Finally we enrich our analysis and introduce a probabilistic model based on partial movement to Boltzmann equilibrium at each stage. This enables us to attach a probability to all partial or final conformations obtained for a particular sequence.

2 METHODS

2.1 Principles

Designing sequences We use designing HP sequences in our study. These are sequences with a unique ground state of lowest energy. Irbäck *et al*. (Irbäck and Troein, 2002) present a list of all designing sequences with up to 24 residues. This provides us with reference sequences against which we can test the sequential folding algorithm.

HP Lattice models We use models which fold on a two-dimensional lattice with residues either hydrophobic or polar. They are said to be in contact if they are adjacent in space but not in sequence. The total energy of the chain is determined by the number of contacts in the conformation simulated.

We let n be the number of residues in the full chain. To study the impact of the variation of the chain length, n takes the value 16 or 24. Evidence has been given that such lengths are capable of mimicking relevant protein behaviour (Chan and Dill, 1993; Chan and Dill, 1994; Dill *et al*., 1995; Pande *et al*., 1997; Shakhnovich, 1998).

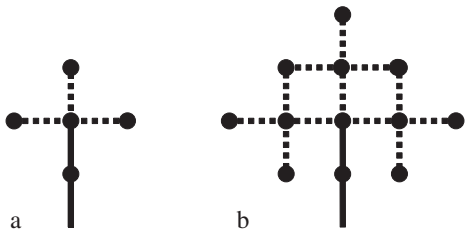


Fig. 2. The different ways to extend a conformation, adding one residue (a) or two (b) at a time. The plain line represents the extremity of the conformation already simulated, and the dashed lines the possible extensions.

Sequential Folding Since we work with relatively short lengths n, the number of monomers s added at each iteration is chosen to be one or two. Sequences of length 16 can have a maximum of nine contacts, so it is reasonable to perform simulations with d, the surmountable energy barrier, equal to zero, one or two.

The first s monomers are laid down, locating them in a conformation with minimum energy, at the same time retaining all configurations within energy d of this minimum. We then have a first set of local conformations of length s and proceed to expand these by adding s monomers to all of these partial configurations, retaining those with minimum energy and all within energy d of this new local minimum. Parameter d remains the surmountable energy barrier, so leading to a new set of local conformations of length $2s$. This procedure is repeated until all monomers are used. A configuration with minimum final energy is termed a ''sequential ground state'', and the one of lowest energy the ''global ground state''.

A conformation C_l, of length l, is extended by s residues using s steps of the three possible single step directions (Figure 2). These three possible directions are—in relative moves—forward, left and right. Only conformations which are self-avoiding and non equivalent are retained. Two conformations are deemed equivalent if one can be obtained either by rotation or reflection on the lattice from the other. At each step we obtain a maximum of three new conformations of length $l+s$. The process is then repeated with each one of these conformations of length $l+s$, and so on until we generate conformations of length n. If n is not a multiple of s, then the algorithm is run for $\lfloor n/s \rfloor$ steps; the last iteration handles the remaining residues.

2.2 Measures of fold compactness

As explained in the introduction, we wish to study the impact of folding sequentially, considering the surmountable energy barrier d, the number of residues added at each iteration of the algorithm s and length of the polypeptide chain n. To assess the final fold we use several measures.

Radius of gyration We calculate the radius of gyration of conformations, as used in real protein structure prediction (Rohl *et al*., 2004; Simons *et al*., 1997; Simons *et al*., 1999), using

$$R_g = \sqrt{\sum_{i=1}^{n} \sum_{j\langle i} [(x_{i1}-x_{j1})^2 + (xi_2-xj_2)^2]}$$

where $x_i = (x_{i1}, x_{i2})$ represents the two coordinates of point i and n is the number of residues in the conformation.

Moment of inertia We use the moment of inertia (MI) as an indicator of the compactness of the structure. It reflects the variance of distances from residues to the centre of mass of the conformation,

$$MI = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2 = \frac{1}{n}\sum_{i=1}^{n}[(x_{i1}-\mu_1)^2 + (xi_2-\mu_2)^2]$$

where $\mu = (\mu_1, \mu_2)$ with $\mu_1 = \frac{1}{n}\sum_{i=1}^{n} x_{i1}$ and $\mu_2 = \frac{1}{n}\sum_{i=1}^{n} xi_2$

We also use a MI restricted to hydrophobic residues. In this case we term the result the hydrophobic moment of inertia (HMI).

Contact signature We define the contact signature S of a conformation to be the average distance in sequence between two residues in contact. So we have

$$S = \frac{\sum_{i<j} d(i,j)\Delta(i,j)}{N_{contacts}}$$

where $d(i,j)=j-i$ is the distance in sequence between the residues at position i and position j and $\Delta(i,j)$ equals one if residues i and j are in contact and zero otherwise; $N_{contacts}$ is the number of contacts in the chain.

3 RESULTS

We use HP models to investigate the difference between the minimum energy state of a controlled sequential folding and the globally minimum energy state. A difference in these two end states will be found if nature is incapable of pushing a partially formed protein back over a sufficiently high free energy barrier. We explore the influence on this difference of n, d and s.

For a particular sequence, the number of final sequential conformations at the minimum energy level decreases as the surmountable energy barrier increases We focus on 149 randomly selected designing sequences of length 16 whose unique global ground state is known. We extrude one residue at a time, so s is equal to one. We first set the surmountable energy barrier d at zero. We observe that for 48 sequences (32.2%) we obtain a unique sequential ground state, which is not necessarily the global ground state. The number of sequences with a unique sequential ground state increases to 95 (63.8%) as we raise d to one. These results suggest that for a given sequence, the number of final conformations decreases as the surmountable energy barrier increases.

As we increase d, the number of local conformations (as described in methods) retained at each step of the elongation increases. Those which are kept have an energy within d of the lowest. If more conformations are simulated, the probability of retaining the global ground state of energy rises. With a surmountable energy barrier sufficiently high, it is possible to enumerate all conformations and then be sure of obtaining the global ground state. Increasing the number of residues extruded at a time has a similar effect. Adding more than one residue at a time increases the number of intermediate conformations simulated as well as the odds of retaining the global ground state.

For the sequence HPPPPHPPPHPHPPHH, for example, a surmountable energy barrier of one is sufficient to access the global state (Figure 3).

Conformations become tighter as the surmountable energy barrier increases Given a particular sequence there are many final sequential folds (with the same energy) for a given s and d. We measure the compactness of the structure with the radius of gyration R_g. We determine the average R_g over all such conformations sequentially generated for a particular sequence. As d increases, the average R_g decreases. We find that for 88% of the sequences, this average R_g remains the same or registers a decrease when we increase d from zero to one, with s equal to one. An example is given in Figure 4.

We also evaluate an average hydrophobic moment of inertia (HMI) of all sequential ground states obtained for a particular

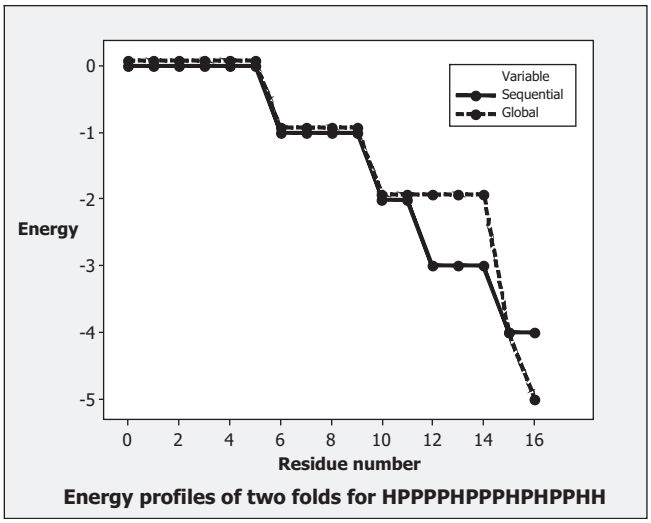


Fig. 3. The energy, in units of $-\varepsilon$, is plotted against the number of residues in the sequence as the chain elongates. The solid line represents the sequential energy path (with d=0 and s=1). The dashed line represents the energy path of the global ground state; note that this path is not influenced by s or d. We observe that the global ground state path is eliminated from the pool of sequential local conformations when the 12th residue is added. At this point the sequential algorithm produces partially extruded conformations with lower energy (one contact). So in this case a surmountable energy barrier of one would be sufficient to retain the path leading to the ground state.

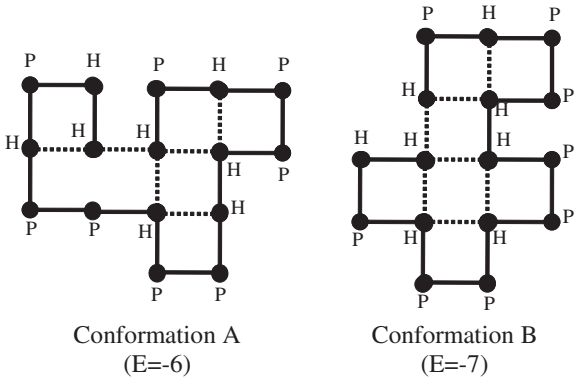


Fig. 4. The graphics show, for the sequence HPHPPHHPPHPPHPHH, the sequential ground state simulated with a surmountable energy barrier equal to zero (left), and equal to one (right). The radius of gyration decreases from 2.405 (left) to 2.377 (right). Both simulations led to a unique sequential ground state.

sequence. We find that the hydrophobic core forms as the surmountable energy barrier increases. We then calculate the difference between this average HMI and the HMI of the global ground state of minimum energy. We observe that as d increases the average HMI of the sequential ground states simulated moves closer to the global HMI. We observe that as d increases, the energy level of final conformations simulated tends to be closer to the energy level of the unique global ground state. The global ground state has the maximum number of contacts possible; hence it generally also has the tightest hydrophobic core. So the closer the conformations are to the

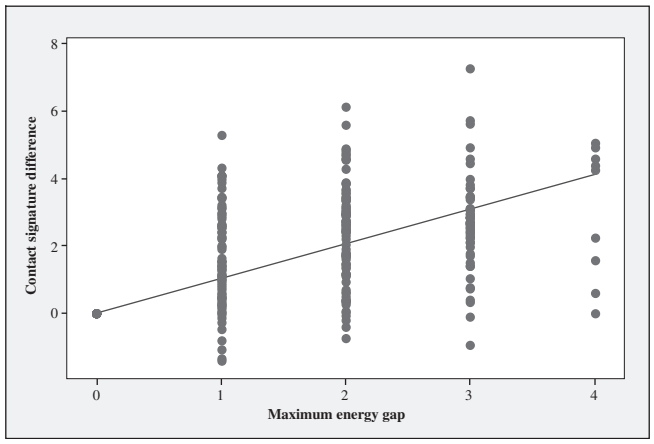


Fig. 5. We evaluate the difference between the average of the sequential contact signatures and the global contact signature. We also determine the maximum energy gap (in the energy paths) between the sequential path of lowest energy and that of the global ground state. The signature difference is plotted against the largest gap in energy. The superimposed points with a null energy gap and a null contact signature difference correspond to the cases where the final sequential conformation is always the global one.

global ground state, the tighter their hydrophobic core is likely to be. In all of the 149 sequences simulated, we observe that 65% (97) have an average HMI which decreases when we increase d from zero to one, and 19.5% (29) have an average HMI which remains the same.

Sequentiality favours short range contacts The further the energy of the sequential path is from that of the global path, the more localized the contacts become. We randomly select 242 sequences of 24 residues and run simulations with d equal to zero and s equal to one. For each sequence we then evaluate the average of the sequential contact signatures, and calculate the difference with the global contact signature. We find that in 89.7% of the cases, the average sequential contact signature is less than the global. We also notice a positive relation when we plot the biggest energy gap for each sequence against the difference in contact signature (Figure 5). These results confirm a previous study which showed that cotranslationality favours local contacts (Morrissey *et al*., 2004).

Some sequences are not foldable sequentially with a low surmountable energy barrier The method explores exhaustively all possible conformations accessible sequentially. Some particular sets of intermediate conformations may result in non-extendable conformations. These are conformations which have folded into a state that cannot be extended to reach the full length conformation. It is possible to avoid these dead-end conformations by increasing the surmountable energy barrier. An increase in d permits a higher number of intermediate conformations to be retained at each iteration of the elongation, and thus reduces the chance that an iteration results only in conformations which cannot be extended. We assume that these conformations which cannot be modelled sequentially with a low surmountable energy barrier cannot represent proteins which have mutated through evolution. We conclude that biological sequences must evolve to avoid sequences which can fall into such traps.

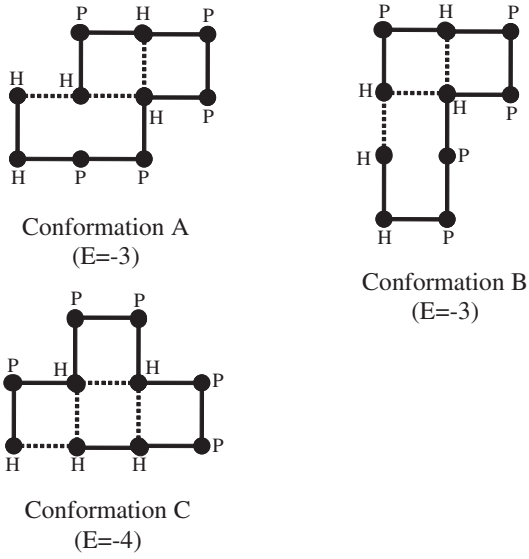


Fig. 6. Conformations A and B represent respectively two (out of eight) sequential ground states with three contacts each, and conformation C shows the unique global ground state for the sequence.

Analysis of energetic pathways of sequentially folded proteins We focus on the 10-mer HPHPPHPPHH. We simulate the ground states obtained with a surmountable energy barrier d=0, adding one residue at a time, so s=1. This sequence corresponds to the shortest designing sequence available for which the unique global ground state fold differs from the global ground state under the preceding conditions. Figure 6 shows two of the conformations obtained sequentially and that of the global ground state. The global ground state can only be reached with a surmountable energy barrier of one.

Figure 7 shows the energy paths of the sequential ground states and the global ground state. When the sixth residue is added, the best fold modelled sequentially has one more contact than the path towards the global ground state of energy. Since the surmountable energy barrier is zero, the path to the global ground state is not retained. Having a null probability of occurrence, the ground state is eliminated from the pool of potential final folds (Figure 8).

Definition of a probabilistic 2D simple lattice model The surmountable energy barrier allows a set of conformations to be retained at each elongation of the chain, and these may have different energies. As a consequence, there may also be a set of final conformations for a given sequence. We want to be able to attach a probability to each of these conformations, partially or fully elongated.

We know that some proteins in their native state are not in their lowest Gibbs free energy state, and fold to a state more stable than the native one (Baker, 1998; Sohl *et al*., 1998). Baskakov *et al*. showed for instance that the folding of mouse prion protein was under kinetic control when folding to its α-helical native conformation, separated by a large energy barrier from a more thermodynamically stable β-sheet-rich isoform (Baskakov *et al*., 2001). Therefore we accept that the intermediate conformations accessed by the polypeptide, as it is elongated, may also not be in a lowest free energy state. In order to model this we do not permit the

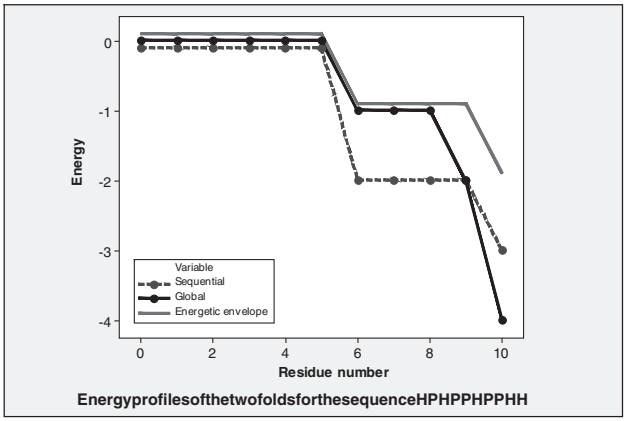


Fig. 7. A plot of the common energy path of the minimum energy sequential folds (with d=0 and s=1) and the global fold for the sequence HPHPPHPPHH. Also shown is the upper energy envelope; all energy paths lying below this envelope are considered in the analysis.

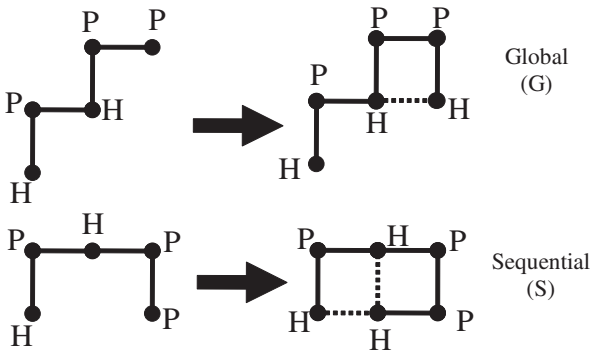


Fig. 8. Conformation G with one contact leads to the global state of energy (for the full length). Conformation G is one of the seven possible conformations of length six with one contact. Conformation S shows the only possible conformation of length six with two contacts. Conformation S is the intermediate conformation of length six which has the lowest energy.

distribution of conformations to reach the Boltzmann energy distribution completely and we introduce a ''thermodynamic permission factor'' β ($0 \leq \beta \leq 1$). This factor is a coefficient permitting movement to the Boltzmann equilibrium probability of every conformation, partially or fully extended.

We now model the probabilities of intermediate conformations along the different energy pathways. The probabilistic model defines a distribution for each intermediate and final model which is the sum of two components, an initial probability weighted by *1-*β and the Boltzmann probability weighted by β. The initial probability is the parent conformation probability divided by the number of offspring of this parent conformation, so is determined by the different elongation paths. If several conformations, after elongation, result in the same offspring conformation, the latter has a chance of occurrence which is the sum of the probabilities of the common offspring. As we assume that the pool of intermediate conformations may not reach the Boltzmann equilibrium, the Boltzmann equilibrium distribution is weighted by β.

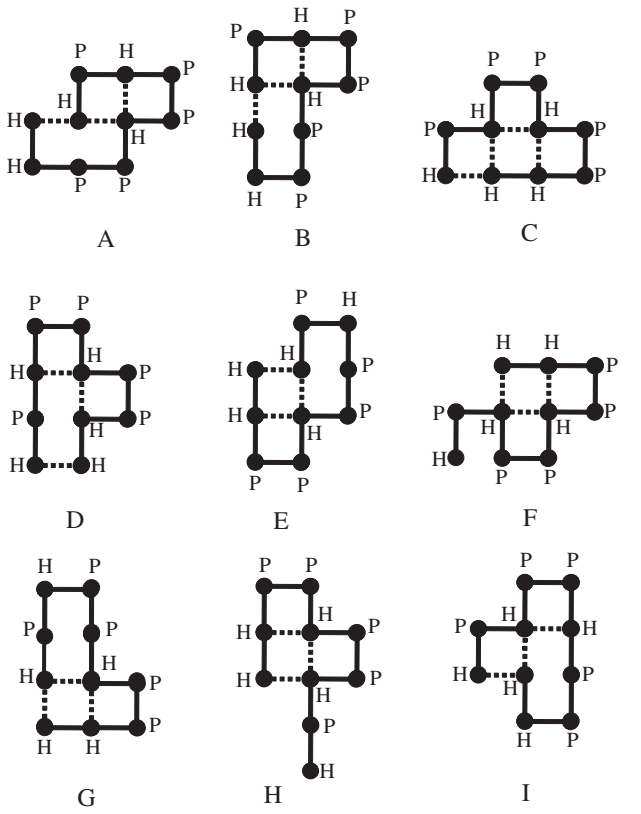


Fig. 9. The nine final conformations obtained sequentially for the sequence HPHPPHPPHH using *s=1* and *d=1*.

Given a surmountable energy barrier d we have, for a chain of l residues, a known distribution of n^l intermediate conformations C_i^l, $i=1,\ldots,n^l$ with known probabilities p_i^l. We elongate all intermediate conformations of length l by s residues. There arises a new set of n^{l+s} intermediate conformations of length $l+s$.

We assume that all newly modeled C_i^{l+s}conformations of length $l+s$ have an immediate probability I_i^{l+s} which is followed in time by a final probability F_i^{l+s}. We know that a given conformation C_i^l can give birth to a number b_i^l of kinetically permissible different conformations of length $l+s$, and that a given conformation C_i^{l+s} can have a_i^{l+s}different ancestors of length l.

We define the initial probability of C_i^{l+s} which has $a_i^{l+s}=a$ ancestors $C_{i1}^l, C_{i2}^l, \ldots, C_{ia}^l$ by

$$I_i^{l+s} = \sum_{j=1}^{a} \frac{F_{ij}^l}{b_{ij}^l}$$

We define the final probability of C_i^{l+s} by

$$F_i^{l+s} = (1-\beta) \times I_i^{l+s} + \beta \times \frac{e^{E_i^{l+s}/kT}}{Q^{l+s}}$$

where E_i^{l+s} is the number of contacts of C_i^{l+s} and

$$Q^{l+s} = \sum_{h=0}^{cl+s} g_{l+s}(h) e^{-h\varepsilon/kT}$$

Table 1. The probability of the nine folds obtained for HPHPPHPPHH

Configuration	Energy	Prob. T=0.2, β=0.75	Prob. T=0.2, β=0.25	Prob. T=0.8, β=0.25
A	−3	0.058	0.274	0.2
B	−3	0.048	0.196	0.152
C	−4	0.737	0.281	0.139
D	−3	0.03	0.05	0.093
E	−3	0.03	0.048	0.089
F	−3	0.03	0.048	0.089
G	−3	0.03	0.048	0.087
H	−3	0.03	0.048	0.087
I	−3	0.005	0.008	0.06

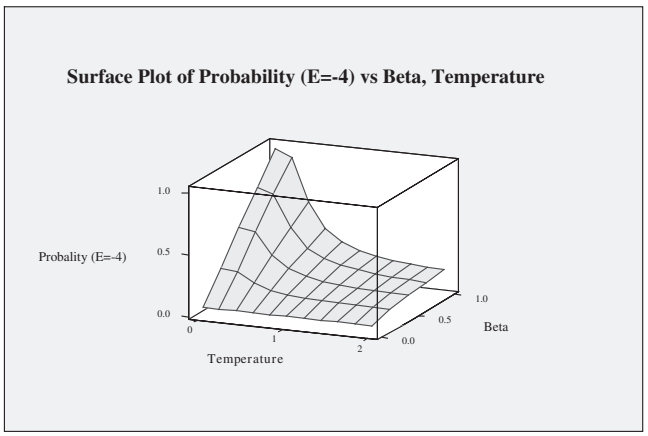


Fig. 10. A graphic showing the probability that the 10-mer HPHPPHPPHH is in the lowest state of energy (−4) as temperature and thermodynamic permission factor change. We observe that the probability decreases as the temperature rises and as the thermodynamic permission factor β drops. When we increase the temperature we allow more energy for unfolding, favouring states which have a higher energy than the ground state. As β decreases to zero, we allow the distribution at each elongation stage less freedom to settle to the Boltzmann distribution, favouring higher energy states. Note that a β of zero results in a model which is independent of temperature, whence the non-zero probability of a final conformation is solely determined by the initial probabilities at each stage.

where Q^{l+s} is the partition function and $g_{l+s}(h)$ is the density of states, which is the number of all sequential conformations of length $l+s$ with h contacts, c_{l+s} is the maximum number of contacts among all conformations of length $l+s$, T is the temperature and k is the Boltzmann constant.

Application of the probabilistic model We apply the probabilistic model to the 10-mer HPHPPHPPHH. We study the impact of β and the temperature T on the distribution of conformations at each step of the elongation process, using $d=1$ and $s=1$. Figure 9 (A-I) shows the nine final conformations obtained; Table 1 shows the final probabilities of these nine conformations. We see that the probability of being in the lowest state of energy (conformation C) decreases as we raise T and lower β. With $T=0.8$ and $\beta=0.25$

Table 2. Summary of biophysical principles modelled

Biophysical mechanisms	Examples	Corresponding parameters in computational experiments	Results in computational experiments	Qualitative prediction to be tested on computational experiments
Cotranslational folding occurs	Semliki Forest virus capsid protein becomes biologically active before the full length polyprotein is produced (Baldwin, 1999)	*s* symbolizes the number of residue(s) added each time the chain is elongated	Models become more compact and less numerous as *s* increases	Results are kinetically controlled folds. Evidence of real protein models to be in kinetic traps is expected (if simulated sequentially under kinetic control)
Folding is under kinetic control	Mouse prion protein native conformation is not the most thermodynamically stable conformation (Baskakov *et al*., 2001)	*d* symbolizes the finite surmountable energy barrier β symbolizes the thermodynamic permission factor which releases the Boltzmann energy distribution	Models become more compact and less numerous as *d* increases Models with highest probability of occurrence are not always the ones of lowest energy	

we have conformations A and B more likely to occur than the lowest energy conformation C. Figure 10 shows the probability that the 10-mer HPHPPHPPHH is in the lowest energy state as T and β vary.

Consequences of cotranslational folding of real proteins Should cotranslational folding prove to be the norm, then we can make predictions about the effect on protein structure:

(i) The N-terminus may be more likely to be buried; the C-terminus, being ''held'' by the ribosome, may be more likely to be peripheral in the final structure.

(ii) Protein structure may favour local contacts.

(iii) The active state of a protein may not be the lowest energy state.

(iv) Designed sequences may often fail to produce the desired structure because cotranslational folding is not taken into account. Therefore designing artificial proteins with local interactions vectorised from the N- to the C- terminus may be advantageous.

(v) New folds of lower energy may be found if we relax kinetic control, increasing the surmountable energy barrier.

CONCLUSION

We have modelled the folding of proteins cotranslationally and under kinetic control, with the help of simple lattice models. We selected intermediate conformations, within the surmountable energy barrier, as the polypeptide chain elongated. We saw that the globally minimum energy, that with the maximum number of contacts, was not always accessible with a low surmountable energy barrier. As we increased this barrier, we obtained final sequential conformations which were more compact and less numerous. A sufficiently high barrier enabled us to reach a final conformation which had the maximum number of contacts.

We attached a probability to each of the intermediate and final folds obtained. We introduced a thermodynamic permission factor, capturing the property that intermediate and final conformations under constraints may not always reach the Boltzmann equilibrium. We found that folds with lowest energy were not always the ones with highest probability. We summarized our results in Table 2.

The study is restricted to short, two-dimensional designing sequences. Modelling could be improved through use of longer sequences, folding three-dimensionally. The thermodynamic permission factor modelled various *in vivo* constraints on the folds, summarizing these constraints in a single parameter. Future developments could include use of a length-dependent thermodynamic permission factor. Finally, we know that the ribosome imposes spatial restrictions on the fold; these should also be taken into account.

REFERENCES

Andersson,S.G.E. and Kurland,C.G. (1990) Codon preferences in free-living microorganisms. *Microbiological Reviews*, **54**, 198–210.

Anfinsen,C.B. (1973) Principles that govern the folding of protein chains. *Science*, **181**, 223–230.

Baker,D. (1998) Metastable states and folding free energy barriers. *Nature Structural Biology*, **5**, 1021–1024.

Baldwin,T.O. (1999) Protein folding in vivo: the importance of ribosomes. *Nature Cell Biology*, **1**, 154–155.

Ban,N., Nissen,P., Hansen,J., Moore,P.B. and Steitz,T.A. (2000) The complete atomic structure of the large ribosomal subunit at 2.4 A resolution. *Science*, **289**, 905–920.

Basharov,M.A. (2003) Protein folding. *Journal of Cellular and Molecular Medicine*, **7**, 223–237.

Baskakov,I.V., Legname,G., Prusiner,S.B. and Cohen,F.E. (2001) Folding of prion protein to its native alpha-helical conformation is under kinetic control. *Journal of Biological Chemistry*, **276**, 19687–19690.

Bornberg-Bauer,E. (1997) Chain growth algorithms for HP-type lattice proteins. *RECOMB 97*, 47–55.

Braakman,I., Hoover-Litty,H., Wagner,K.R. and Helenius,A. (1991) Folding of influenza hemagglutinin in the endoplasmic reticulum. *Journal of Cell Biology*, **114**, 401–411.

Bujnicki,J.M. (2006) Protein-structure prediction by recombination of fragments. *ChemBioChem*, **7**, 19–27.

Chan,H.S. and Dill,K.A. (1993) Energy landscapes and the collapse dynamics of homopolymers. *Journal of Chemical Physics*, **99**, 2116–2127.

Chan,H.S. and Dill,K.A. (1994) Transition states and folding dynamics of proteins and heteropolymers. *Journal of Chemical Physics*, **100**, 9238–9257.

Curran,J.F. and Yarus,M. (1989) Rates of aminoacyl-tRNA selection at 29 sense codons in vivo. *Journal of Molecular Biology*, **209**, 65–77.

Dill,K.A., Bromberg,S., Yue,K., Fiebig,K.M., Yee,D.P., Thomas,P.D. and Chan,H.S. (1995) Principles of protein folding—A perspective from simple exact models. *Protein Science*, **4**, 561–602.

Fedorov,A.N. and Baldwin,T.O. (1997) Cotranslational protein folding. *Journal of Biological Chemistry*, **272**, 32715–32718.

Fedorov,A.N. and Baldwin,T.O. (1997) GroE modulates kinetic partitioning of folding intermediates between alternative states to maximize the yield of biologically active protein. *Journal of Molecular Biology*, **268**, 712–723.

Fernandez,A. (1994) Ascribing weights to folding histories: explaining the expediency of biopolymer folding. *Journal of Physics A (Mathematical and General)*, **27**, 6039–6052.

Frydman,J. (2001) Folding of newly translated proteins in vivo: the role of molecular chaperones. *Annual Review of Biochemistry*, **70**, 603–647.

Govindarajan,S. and Goldstein,R.A. (1998) On the thermodynamic hypothesis of protein folding. *Proceedings of the National Academy of Sciences*, **95**, 5545–5549.

Guo,Z., Brooks,C.L. and Boczko,E.M. (1997) Exploring the folding free energy surface of a three-helix bundle protein. *Proceedings of the National Academy of Sciences*, **94**, 10161–10166.

Hartl,F.U. and Hayer-Hartl,M. (2002) Molecular chaperones in the cytosol: from nascent chain to folded protein. *Science*, **295**, 1852–1858.

Irbäck,A. and Troein,C. (2002) Enumerating designing sequences in the HP model. *Journal of Biological Physics*, **28**, 1–15.

Jenni,S. and Bany,N. (2003) The chemistry of protein synthesis and voyage through the ribosomal tunnel. *Current Opinion in Structural Biology*, **13**, 212–219.

Keller,S.L. (2003) Sequential folding of a rigid wire into three-dimensional structures. *American Journal of Physics*, **72**, 599–604.

Kolb,V.A. (2001) Cotranslational protein folding. *Molecular Biology*, **35**, 584–590.

Kolb,V.A., Makeyev,E.V. and Spirin,A.S. (2000) Co-translational folding of an eukaryotic multidomain protein in a prokaryotic translation system. *Journal of Biological Chemistry*, **275**, 16597–16601.

Komar,A.A. and Jaenicke,R. (1995) Kinetics of translation of gamma B crystallin and its circularly permutated variant in an in vitro cell-free system: possible relations to codon distribution and protein folding. *FEBS Letters*, **376**, 195–198.

Komar,A.A., Lesnik,T. and Reiss,C. (1999) Synonymous codon substitutions affect ribosome traffic and protein folding during in vitro translation. *FEBS Letters*, **462**, 387–391.

Levinthal,C. (1968) Are there pathways for protein folding, *Journal of Chemical Physics*, **65**, 44–45.

Levinthal,C. (1969) Mossbauer spectroscopy in biological systems. *University of Illinois Press, Urbana*, 22–24.

Morrissey,M.P., Ahmed,Z. and Shakhnovich,E.I. (2004) The role of cotranslation in protein folding: a lattice model study. *Polymer*, **45**, 557–571.

Nakatogawa,H. and Ito,K. (2002) The ribosomal exit tunnel functions as a discriminating gate. *Cell*, **106**, 629–636.

Netzer,W.J. and Hartl,F.U. (1997) Recombination of protein domains facilitated by co-translational folding in eukaryotes. *Nature*, **388**, 343–349.

Nicola,A.V., Chen,W. and Helenius,A. (1999) Co-translational folding of an alphavirus capsid protein in the cytosol of living cells. *Nature Cell Biology*, **1**, 341–345.

Pande,V.S., Grosbergy,A.Y. and Tanaka,T. (1997) Statistical mechanics of simple models of protein folding and design. *Biophysical Journal*, **73**, 3192–3210.

Purvis,I.J., Bettany,A.J.E., Santiago,T.C., Coggins,J.R., Duncan,K., Eason,R. and Brown,A.J.P. (1987) The efficiency of folding of some proteins is increased by controlled rates of translation in vivo. A hypothesis, *Journal of Molecular Biology*, **193**, 413–417.

Ramakrishnan,V. (2002) Ribosome structure and the mechanism of translation. *Cell*, **108**, 557–572.

Rohl,C.A., Strauss,C.E.M., Misura,K.M.S. and Baker,D. (2004) Protein structure prediction using Rosetta. *Methods Enzymol*, **383**, 66–93.

Sanchez,I.E., Morillas,M., Zobeley,E., Kiefhaber,T. and Glockshuber,R. (2004) Fast folding of the two-domain semliki forest virus capsid protein explains co-translational proteolytic activity. *Journal of Molecular Biology*, **338**, 159–167.

Shakhnovich,E.I. (1998) Protein design: A perspective from simple tractable models. *ArXiv Condensed Matter e-prints/9804199 (web publication)*.

Sikorski,A. and Skolnick,J. (1990) Dynamic monte carlo simulations of globular protein folding. *Journal of Molecular Biology*, **215**, 183–198.

Simons,K.T., Kooperberg,C., Huang,E. and Baker,D. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *Journal of Molecular Biology*, **268**, 209–225.

Simons,K.T., Ruczinski,I., Kooperberg,C., Fox,B.A., Bystroff,C. and Baker,D. (1999) Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. *Proteins*, **34**, 82–95.

Sohl,J.L., Jaswal,S.S. and Agard,D.A. (1998) Unfolded conformations of alpha-lytic protease are more stable than its native state. *Nature*, **392**, 817–819.

Wilson,D.N., Blaha,G., Connell,S.R., Ivanov,P.V., Jenke,H., Stelzl,U., Teraoka,Y. and Nierhaus,K.H. (2002) Protein synthesis at atomic resolution: mechanistics of translation in the light of highly resolved structures for the ribosome. *Current Protein and Peptide Science*, **3**, 1–53.

Ziv,G., Haran,G. and Thirumalai,D. (2005) Ribosome exit tunnel can entropically stabilize alpha-helices. *Proceedings of the National Academy of Sciences*, **102**, 18956–18961.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e211–e219
doi:10.1093/bioinformatics/btl233

BNTagger: improved tagging SNP selection using Bayesian networks

Phil Hyoun Lee* and Hagit Shatkay*

School of Computing, Queen's University, Kingston, ON, Canada

ABSTRACT

Genetic variation analysis holds much promise as a basis for disease-gene association. However, due to the tremendous number of candidate single nucleotide polymorphisms (SNPs), there is a clear need to expedite genotyping by selecting and considering only a subset of all SNPs. This process is known as *tagging SNP selection*. Several methods for tagging SNP selection have been proposed, and have shown promising results. However, most of them rely on strong assumptions such as prior block-partitioning, bi-allelic SNPs, or a fixed number or location of tagging SNPs.

We introduce BNTagger, a new method for tagging SNP selection, based on conditional independence among SNPs. Using the formalism of Bayesian networks (BNs), our system aims to select a subset of independent and highly predictive SNPs. Similar to previous prediction-based methods, we aim to maximize the prediction accuracy of tagging SNPs, but unlike them, we neither fix the number nor the location of predictive tagging SNPs, nor require SNPs to be bi-allelic. In addition, for newly-genotyped samples, BNTagger directly uses genotype data as input, while producing as output haplotype data of all SNPs.

Using three public data sets, we compare the prediction performance of our method to that of three state-of-the-art tagging SNP selection methods. The results demonstrate that our method consistently improves upon previous methods in terms of prediction accuracy. Moreover, our method retains its good performance even when a very small number of tagging SNPs are used.

Contact: lee@cs.queensu.ca, shatkay@cs.queensu.ca

1 INTRODUCTION

A major interest of current genomics research is *disease-gene association*, that is, identifying which DNA variations are highly associated with a specific disease. In particular, single nucleotide polymorphisms (SNPs), which are the most common form of DNA variation, as well as sets of SNPs localized on one chromosome—referred to as *haplotypes*—are at the forefront of disease-gene association studies (Halldörsson *et al*., 2004b; Crawford and Nickerson, 2005). However, in most large-scale association studies, genotyping all SNPs in a candidate region for a large number of individuals is still costly and time-consuming. Thus, selecting a subset of SNPs that is sufficiently informative but still small enough to reduce the genotyping overhead is an important step toward disease-gene association. This process is known as *haplotype tagging SNP (htSNP) selection*, and it poses a current major challenge (Crawford and Nickerson, 2005; Johnson *et al*., 2001).

Several computational methods for htSNP selection have been proposed in the past few years. One widely-used approach is based on *the block structure of the human genome* (Daly *et al*., 2001; Gabriel *et al*., 2002). That is, the human genome can be viewed as a set of discrete blocks such that within each block, there is a very small set of common haplotypes shared by most of the population (i.e., 80–90%). Based on this idea, these methods aim to identify a subset of SNPs that can distinguish all the common haplotypes (Gabriel *et al*., 2002), or at least explain a certain percentage of them (Johnson *et al*., 2001; Avi-Itzhak *et al*., 2003). Another popular htSNP selection approach (Ao *et al*., 2005; Carlson *et al*., 2004), rooted in linkage disequilibrium (LD), is based on *pairwise association* of SNPs. This approach tries to select a set of htSNPs such that each of the SNPs on a haplotype is *highly associated* with one of the htSNPs. This way, although the SNP that is directly responsible for the disease may not be selected as an htSNP, the association of the target disease with that SNP can be indirectly deduced from its associated htSNP.

Bafna *et al*. (2003) and Halldörsson *et al*. (2004) proposed a somewhat different approach. They consider htSNPs to be a subset of all SNPs, from which the remaining SNPs can be reconstructed. Thus, they aim to select htSNPs based on how well they *predict* the remaining set of the unselected SNPs, referred to as *tagged* SNPs, and *reconstruct* the complete haplotypes using htSNPs. To quantify the confidence with which one group of SNPs can predict another, they suggested a new measure called *informativeness*. With the same predictive aim, Halperin *et al*. (2005) also proposed a new measure, directly evaluating the prediction accuracy of a set of SNPs. By limiting the number of predictive SNPs or restricting them to a w-bounded neighborhood (where w is a fixed window size ≤ 30), both methods can identify the optimal (under these restrictions) set of htSNPs satisfying their respective figure of merit.

These last two methods are not based on the block structure of the human genome. Thus, they do not assume prior block partitioning or limited diversity of haplotypes. Furthermore, they can use a combination of several SNPs to predict the others. Therefore, predictive methods typically select a smaller number of htSNPs than pairwise association methods (De Bakker *et al*., 2006). However, despite their advantages, these predictive methods still suffer from several limitations. All of them can only be applied to bi-allelic SNPs (i.e., ones

*To whom correspondence should be addressed.

having only two different alleles[1]), and their performance is limited by restrictions such as the small-bounded location or the fixed number of htSNPs for each prediction. In addition, most of them require haplotype information of htSNPs to reconstruct newly-genotyped samples.

In this paper, we present a new method, BNTagger, for selecting htSNPs based on their accuracy in predicting tagged SNPs, that is not limited by previous restrictions. In addition, we provide a haplotype-reconstruction framework for newly-genotyped samples. To identify a predictor-predictee relationship among SNPs, we utilize conditional independencies among SNPs in the framework of Bayesian networks. Bayesian networks (BNs) have been previously used for haplotype block partitioning (Greenspan and Geiger, 2003) and haplotype phasing (Xing *et al*., 2004), but to our knowledge, this is the first time that they are applied to htSNP selection. BNTagger uses three main steps:

(1) Identifying the conditional independence relations among SNPs.

(2) Selecting htSNPs using two heuristics.

(3) Reconstructing the complete haplotypes for newly-genotyped samples.

Similar to other predictive methods, our system aims to select htSNPs maximizing the prediction accuracy for the remaining tagged SNPs. However, it has several unique aspects. First, unlike all previous work (Bafna *et al*., 2003; Halldörsson *et al*., 2004; Halperin *et al*., 2005), we do not fix the neighborhood nor the number of predictive htSNPs for each tagged SNP. Although SNPs within close physical proximity are assumed to be in a state of high linkage disequilibrium (LD), recent studies have reported that the levels of LD vary across chromosomal regions (Reich *et al*., 2001; Daly *et al*., 2001). Therefore, as noted by Bafna *et al*. (2003), *''... it is neither efficient nor desirable to fix the neighborhood in which htSNPs are selected''*. Moreover, it is realistic to assume that a different number of htSNPs may be needed for predicting each tagged SNP.

Second, our system is not restricted to the case of bi-allelic SNPs. While most SNPs are indeed bi-allelic, there are SNPs that can take on more than two nucleotides. While these cases may be rare, it is still unknown whether disease variants are rare or common haplotypes (Crawford and Nickerson, 2005). Thus, it is desirable to impose as few restrictions as possible on htSNP selection (Palmer and Cardon, 2005).

Third, for newly-geneotyped samples, we directly construct *haplotype* data of *all* SNPs using *genotype* data of htSNPs. As pointed by Halperin *et al*. (2005), the accuracy of haplotype phasing based only on htSNPs is limited due to the reduced LD among htSNPs. Therefore, it is reasonable to assume that reliable haplotype data are not available in the case of newly-genotyped samples. However, we note that, unlike Halperin's method, which uses genotype data as input and as output as well, we directly output the *haplotype* data of all SNPs for new samples. Thus, subsequent haplotype phasing for the reconstructed samples is unnecessary.

We applied our method to three public data sets (Daly *et al*., 2001; Rieder *et al*., 1999; Nickerson *et al*., 2000). Based on leave-one-out and on 10-fold cross validation, our results demonstrate that using our selection method, about 2.9%–11.5% of the total SNPs are sufficient to predict the others with 90% accuracy. We also compare our prediction performance to that of recently published htSNP selection methods (Bafna *et al*., 2003; Halldörsson *et al*., 2004; Lin and Altman, 2004; Halperin *et al*., 2005). The results show that our method extracts *fewer htSNPs* while achieving the same level of prediction accuracy. Moreover, our method retains its good performance even when a very small number of htSNPs is used.

In section 2, we formulate the problem of htSNP selection in the context of prediction accuracy, and introduce the basic notations that are used throughout the paper. Section 3 briefly provides the necessary background on Bayesian networks, focusing on the concepts most relevant to our algorithm. Our selection and haplotype reconstruction algorithms are described in section 4. Section 5 reports our evaluation results. Section 6 summarizes our findings and outlines future directions.

2 PROBLEM FORMULATION

A haplotype represents the allele information of contiguous SNPs on *one* chromosome, while a genotype represents the *combined* allele information of the SNPs on a *pair* of chromosomes. Thus, the allele information of haplotypes takes on values from $\{a, g, c, t\}$, while that of genotypes takes on values from $\{a/a, a/g, a/c, a/t, \ldots, t/c, t/t\}$. When the combined allele information of a pair of haplotypes, h_j and h_k, comprises the genotype g_i, we say that h_j and h_k *resolve* g_i. For example, the two haplotypes $h_j = (a, g, a, c)$ and $h_k = (a, c, c, a)$ resolve the genotype $g_i = (a/a, c/g, a/c, a/c)$. We also refer to haplotypes h_j and h_k as the *complementary mates* of each other to resolve g_i, and consider them to be *compatible* with g_i.

Let D be a data set consisting of n haplotypes, $h_1, \ldots, h_n$, each with p different SNPs, $s_1, \ldots, s_p$. The set D can be viewed as an n by p matrix. Each row, D_{i-}, in D corresponds to haplotype h_i, while each column, D_{-j}, corresponds to a SNP s_j. D_{ij} denotes the j^{th} SNP in the i^{th} haplotype. We view each SNP as a discrete random variable, X_j, that takes on values from a finite domain $\{a, g, c, t\}$. Thus, we define the finite set $V = \{X_1, \ldots, X_p\}$, in which each random variable X_j corresponds to the j^{th} SNP on a haplotype in the data set D.

Given the set V of random variables corresponding to the p SNPs, our goal is to find a subset $T \subset V$, such that the size of T, $|T|$, is smaller than some pre-specified constant k, and SNPs in T can best predict the remaining unselected ones, $V - T$. As defined earlier, the selected SNPs are referred to as haplotype *tagging* SNPs (htSNPs), and the unselected ones are referred to as *tagged* SNPs. Suppose that our htSNP set T consists of q SNPs, $T = \{X_{t_1}, \ldots, X_{t_q}\}$. To predict the allele of a tagged SNP X_j given the alleles of the htSNPs, T, we use the posterior probability of X_j conditioned on the set T, $Pr(X_j \mid X_{t_1}, \ldots X_{t_q})$. That is, the allele whose conditional probability is the highest given the alleles of the predictive htSNPs is taken to be the allele of the tagged SNP. When multiple maximum probability solutions exist, the most common allele of X_j is selected. To capture the idea that this prediction can be either correct or incorrect, we introduce the following indicator function P_f.

[1]The nucleotide $\in \{a, g, c, t\}$ at a position in which a SNP occurred is called an *allele*.

DEFINITION 1. *Prediction Indicator Function: Given a predictive htSNP set, $T = \{X_{t_1}, \ldots, X_{t_q}\}$, a predicted tagged SNP, $X_j \in V - T$, and a haplotype, D_{i-}, a prediction indicator function $P_f(X_j, T, D_{i-})$ is defined*[2] *as*

$$P_f(X_j, T, D_{i-}) = \begin{cases} 1 : \text{if } D_{ij} == \\ \quad \underset{x \in \{a,g,c,t\}}{\arg\max}\ Pr(X_j = x \mid X_{t_1} = D_{it_1}, \ldots, X_{t_q} = D_{it_q}); \\ 0 : \text{otherwise}. \end{cases}$$

We note that the prediction of each tagged SNP is assumed to depend on the values of the htSNPs, but not on the other predicted tagged SNPs. Hence, prediction can be applied in any order. Using this prediction indicator function, we formally define our objective as follows:

DEFINITION 2. *Maximally Predictive htSNP Set: Given a set of p SNPs, $V = \{X_1, \ldots, X_p\}$, a constant k, and a prediction indicator function P_f, a maximally predictive htSNP set, $T = \{X_{t_1}, \ldots, X_{t_q}\}$, for a set of haplotypes D is defined as a subset T of V, $(T \subset V)$, satisfying two criteria*:

$$1) \quad |T| < k, \quad \text{and}$$
$$2) \quad T = \underset{T' \subset V}{\text{argmax}} \sum_{j=1}^{p} \sum_{i=1}^{n} p_f(X_j, T', D_{i-}).$$

That is, T is the subset of SNPs that is likely to predict correctly the largest number of SNPs in $V - T$. BNTagger utilizes the framework of Bayesian networks to effectively compute the posterior probability in P_f and to select a set of htSNPs. In the next section, we briefly introduce the necessary background on Bayesian networks.

3 BAYESIAN NETWORKS

A Bayesian network (BN) is a graphical model of joint probability distributions that captures conditional independencies among its variables (Jensen, 2002). Given a finite set $V = \{X_1, \ldots, X_p\}$ of random variables, a Bayesian network has two components: a directed acyclic graph, G, and a set of conditional probability parameters, $\Theta = \{\theta_1, \ldots, \theta_p\}$. Each node of the graph G corresponds to a random variable X_j. An edge between two nodes represents a direct dependence between the two random variables, and the lack of an edge represents their *conditional independence*. Using the conditional independence encoded in the structure of the BN (Jensen, 2002), the joint probability distribution of the random variables in V can be computed as the product of their conditional probability parameters:

$$Pr(V) = \prod_{j=1}^{p} \theta_j = \prod_{j=1}^{p} Pr(X_j \mid pa(X_j)),$$

where $pa(X_j)$ denotes the *parent* nodes of X_j. The BN formalism enables the computation of the posterior probability of a target variable when the values of some of the other variables are observed. This computation process is typically referred to as *BN inference*. Suppose that we have observed the values of q variables, $X_{t_1} = e_1, \ldots, X_{tq} = e_q$, in a BN. Based on this information, the conditional distribution of X_j can be computed from the joint probability of V by marginalizing out all unobserved variables except X_j, denoted as $M = V - \{X_j, X_{t_1}, \ldots, X_{t_q}\}$ (Jensen, 2002). Let m denote any of the possible instantiation of the random variables in M. The posterior probability of X_j can thus be calculated as:

$$\begin{aligned} &Pr(X_j \mid X_{t_1} = e_1, \ldots, X_{t_q} = e_q) \\ &= \frac{\sum_m Pr(M = m, X_j, X_{t_1} = e_1, \ldots, X_{t_q} = e_q)}{Pr(X_{t_1} = e_1, \ldots, X_{t_q} = e_q)} \\ &= \frac{\sum_m \prod_{X_k \in V} Pr(X_k \mid pa(X_k))^*}{Pr(X_{t_1} = e_1, \ldots, X_{t_q} = e_q)}, \end{aligned} \quad (1)$$

where the summation is over all possible combinations of values m assigned to all the unobserved variables in M, and the value of every observed variable, X_{t_i}, is set to e_i in $Pr(X_k \mid pa(X_k))^*$.

The *Markov blanket* is another central concept in Bayesian networks. The Markov blanket of X_j includes the parents of X_j, the children of X_j, and the other parents of X_j's children (Jensen, 2002). In a BN, X_j is conditionally independent of all other variables given its Markov blanket. This typically speeds up the calculation of the posterior $Pr(X_j \mid X_{t_1} = e_1, \ldots, X_{t_q} = e_q)$ since when the Markov blanket of X_j is observed, only this information needs to be taken into account for computing the distribution of X_j.

Numerous BN inference algorithms have been developed to compute this posterior probability exactly or approximately. We use the *Generalized Variable Elimination* algorithm implemented in JavaBayes (Cozman, 2000) to compute the posterior probability used in our prediction indicator function P_f.

To use the BN inference algorithm, we must first identify the structure (G) and parameters (Θ) of the BN representing the haplotype data D. This process is referred to as *BN learning*. *Structure* learning aims to find the graph structure G which maximizes the conditional probability of G given the data D, as follows:

$$\begin{aligned} G &= \underset{G'}{argmax}\ Pr(G' \mid D) = \underset{G'}{argmax}\ \frac{Pr(D \mid G') \cdot Pr(G')}{Pr(D)} \\ &= \underset{G'}{argmax}\ Pr(D \mid G') \cdot Pr(G'). \end{aligned}$$

We use the Minimum Description Length (MDL) score (Lam and Bacchus, 1994) to reflect the above probabilistic scoring. In the same vein, *parameter* learning in a BN aims to find Θ which maximizes the conditional probability of Θ given the data D, $Pr(\Theta \mid D)$. We use a maximum-likelihood approach to estimate Θ.

4 METHODS

BNTagger aims to select a set of htSNPs that predicts the tagged SNPs with the highest accuracy. However, finding this set of htSNPs in the general case has been proven to be NP-hard (Bafna *et al.*, 2003). To effectively identify the set of highly predictive SNPs, T, we use several heuristics, utilizing the framework of a Bayesian network (BN) and the conditional independence captured in it.

Figure 1 provides a simple example for how BNTagger utilizes the conditional independencies among SNPs to select htSNPs. The sample here consists of ten haplotypes with four SNPs each (Figure 1(a)); the BN structure that represents conditional independencies among the four SNPs along with the probability parameters is found via BN learning, and shown in Figure 1(b). For simplicity, the conditional probabilities are

[2]For any SNP $X_{t_i} \in T$, $P_f(X_{t_i}, T, D_{i-})$ is taken to be 1 always.

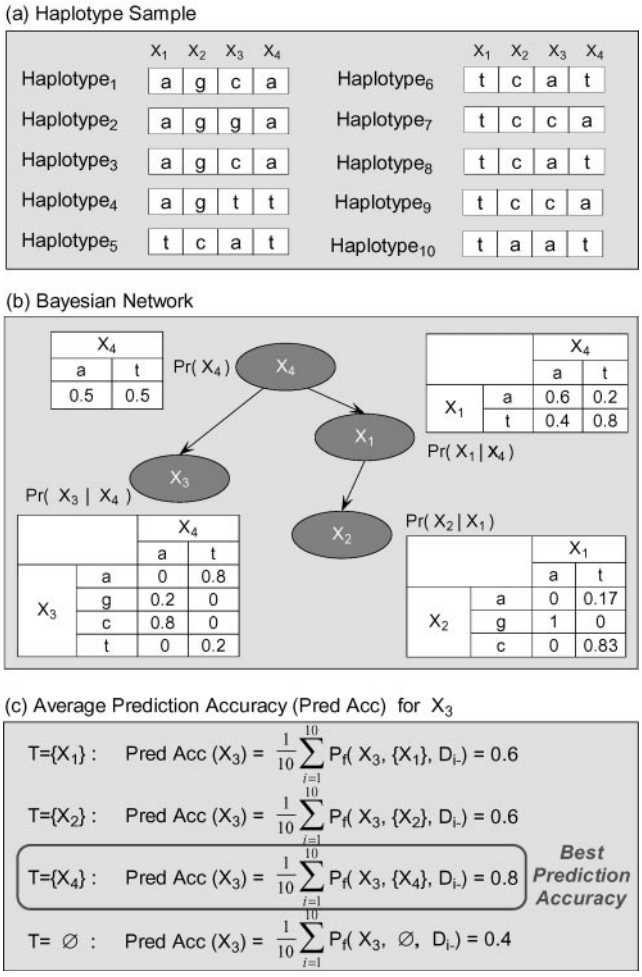


Fig. 1. A Bayesian network of SNPs and examples of prediction accuracy values.

shown only for alleles occurring in the sample. The other probabilities are considered here to be zero.

To select htSNPs given a Bayesian network, BNTagger starts with an empty htSNP set T, and sequentially examines the average prediction accuracy for each SNP (node) based on the current set, T. If the prediction accuracy for a SNP, X_j, is smaller than a pre-specified threshold, BNTagger adds X_j into T as a new htSNP, because X_j is not well-predicted by the current htSNPs in T. Clearly, the order in which SNPs are evaluated is very important, since it can directly affect the selected set of htSNPs and their prediction performance. Unlike other methods that sequentially examine SNPs in the order of their *chromosomal location*, BNTagger examines the SNPs in the *topological* order (from parents to children) in the BN. For example, in Figure 1(b), BNTagger first examines the root X_4, then its children X_3, X_1, and so on. Thus, when the prediction accuracy for each SNP X_j is evaluated, given T, the htSNPs in the current set T are all ancestors of X_j. This has two advantages:

First, the parent-child relation in the BN encodes the direct dependence between these nodes, that is, the state of child nodes depends primarily on the information of their parents. For example, Figure 1(c) shows the prediction accuracy[3] for SNP X_3 assuming each of the other SNPs, X_1, X_2, or X_4 as an htSNP, as well as when assuming no htSNP is used. All the prediction accuracies are higher when htSNP information is given than when it is not. Moreover, the best prediction accuracy is achieved when the parent of X_3, that is X_4, is used as a predictor.

Second, as shown in Definition 1, BNTagger calculates the prediction accuracy for each SNP X_j using the posterior probability of X_j *given* the allele information of the htSNPs. To calculate this posterior, the product of the conditional probabilities in the BN must be computed as was shown in Equation (1). However, if the set of htSNPs contains no descendants of X_j and the parents of X_j are already in the set of htSNPs, the posterior probability is the same as the conditional probability parameter of X_j, due to the conditional independence encoded in the BN. For instance, in Figure 1(c), the best prediction accuracy for the SNP X_3 is simply the maximum of its conditional probability parameters, $Pr(X_3 \mid X_4)$, shown in Figure 1(b).

As a result, the conditional independence structure and the conditional probability parameters in the BN guide BNTagger to find a set of highly predictive htSNPs, and expedite the evaluation procedure. We note though that in order to use the BN components, BNTagger must first build them. Once the BN is constructed and the htSNPs are selected, we also provide a reconstruction framework for newly-genotyped samples; as mentioned earlier, the main purpose of prediction-based htSNP selection is to *reconstruct* the original set of SNP information based on the selected htSNPs.

To summarize, BNTagger consists of three stages: I. Identification of the conditional independence relations among SNPs; II. htSNP selection; and III. Reconstruction of haplotype information for newly-genotyped samples. In the first stage, BN learning is used to identify a graph structure, G, and a set of conditional probability parameters, Θ, that best explain the given haplotype data, D. In the second stage, a heuristic search is applied to the identified BN model to find a set of htSNPs. The third stage provides the haplotype reconstruction framework for subsequent association studies. These three stages are depicted in Figure 2, and are further described in the following subsections.

4.1 Identification of conditional independence relations among SNPs

To use a Bayesian network as described above, its structure and parameters must first be *learned*. We implemented the *Sparse Candidate* algorithm (Friedman *et al*., 1999), which accelerates BN learning by restricting the parents of each node to a small subset of candidates. To select candidate parents for each node, we use the non-random association among SNPs, known as linkage disequilibrium (LD). Disease-gene association studies are typically based on the assumption that LD exists between a disease allele and adjacent SNPs (Crawford and Nickerson, 2005), thus it is widely used for quantifying relationships between SNPs in population genetics. Numerous LD measures have been used. Among them, we use the multi-allelic[4] extension of Lewontin's linkage disequilibrium (LD) measure, D' (Hedrick, 1987), which is one of the most commonly used measures for multi-allelic SNPs (Aulchenko *et al*., 2003).

We explain it here in detail. Let X_1 be an m-allelic SNP, and X_2 be an n-allelic SNP. Let f_i^1 be the relative frequency of the i^{th} allele for SNP X_1, while f_j^2 be the relative frequency of the j^{th} allele for SNP X_2. Let f_{ij} be the relative joint frequency of the i^{th} allele occurring for SNP X_1 and the j^{th} allele occurring for SNP X_2 (where $i = 1, \ldots, m$ and $j = 1, \ldots, n$). Formally, the multi-allelic extension of Lewontin's LD, D', is defined as:

$$D' = \sum_{i=1}^{m} \sum_{j=1}^{n} f_i^1 \cdot f_j^2 \left| \frac{f_{ij} - f_i^1 \cdot f_j^2}{D_{max}} \right|,$$

where D_{max} is the maximum value of LD between the i^{th} and the j^{th} alleles. In principle, D' measures the difference between the observed (f_{ij}) and the

[3]The prediction indicator function P_f (Definition 1) is used in the equations in Figure 1(c).

[4]Most LD measures assume SNPs to have only two different alleles. Multi-allelic LD measures extend these bi-allelic LD measures, by allowing SNPs to have more than two different alleles.

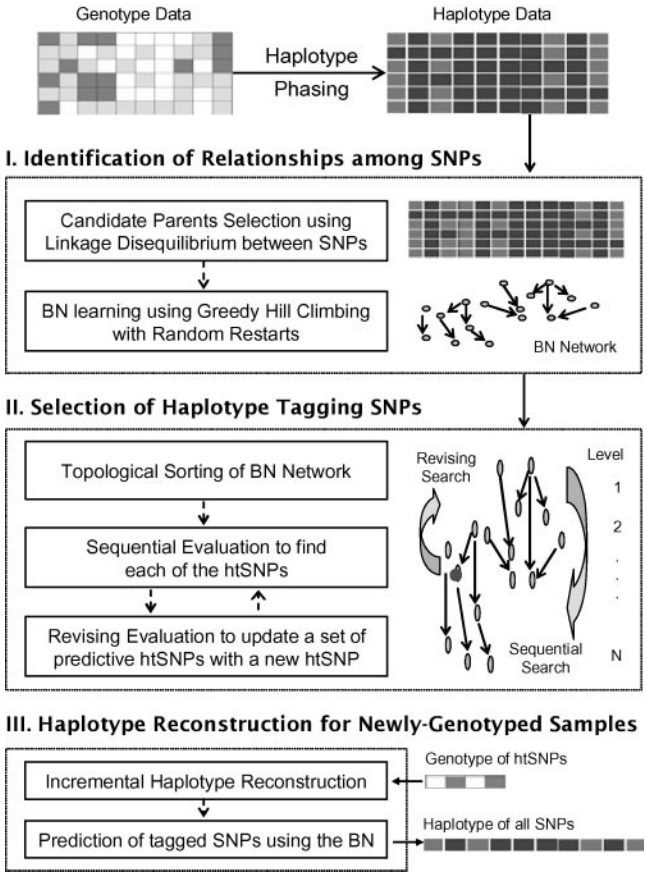


Fig. 2. Outline of haplotype tagging SNP selection and reconstruction in BNTagger.

expected frequency of haplotypes under independence ($f_i^1 \cdot f_j^2$), normalized by the maximum LD (D_{max}), and weighted by the expected joint frequency under independence ($f_i^1 \cdot f_j^2$).

Using the measure D', BNTagger first considers candidate parents for SNP X_j from the set $V - \{X_j\}$, whose pairwise disequilibrium with X_j, as measured by D', is in the top γ percent (here, $\gamma = 10$). The search for the optimal graph structure is performed using greedy hill climbing with random restarts. After N iterations ($N = 25{,}000$), we select the graph structure with the best MDL score (Lam and Bacchus, 1994). The conditional probability parameters $\Theta = \{\theta_1, \ldots, \theta_p\}$ are computed using maximum-likelihood estimation given the identified structure and the data.

4.2 Haplotype tagging SNP selection

Given the SNP-independence structure and the parameters constructed in the previous stage, we now identify a set of htSNPs, T, for the haplotype data, D. Since a different combination of htSNPs can be used to predict each tagged SNP, we also identify a set of predictive htSNPs, $T_{X_j} \subset T$, for each tagged SNP X_j.

As was demonstrated earlier, given the haplotype data, D, and the current set of htSNPs, T, we sequentially examine the average prediction accuracy for each SNP, X_j. If the prediction accuracy for the SNP X_j is smaller than a pre-specified threshold, α, X_j is added to the set of htSNPs, T. Otherwise, X_j is considered a tagged SNP, and the current htSNP set, T, is kept as its *candidate* set of predictive htSNPs, T_{X_j}. We call this procedure *sequential search*. When a new htSNP is added to T during the sequential search, we re-evaluate the prediction accuracy for previously examined tagged SNPs using the updated T. If the prediction accuracy for the re-examined tagged SNP is increased by using the new set T, its previously assigned candidate set of predictive htSNPs is updated to the new T. We call this procedure *revising search*.

In brief, BNTagger sequentially identifies a global set of htSNPs, T, based on their prediction accuracy, and iteratively updates the predictive set of htSNPs, T_{X_j}, for each tagged SNP, X_j. To efficiently conduct these procedures, BNTagger uses two heuristics. First, we topologically sort the nodes in the BN, which yields the *levels* of nodes as defined below, and conduct sequential search in this topological order.

DEFINITION 3. *A level of node X_j in a Bayesian network is defined as*:

$$level(X_j) = \begin{cases} 1 & : \text{if } pa(X_j) = \phi; \\ \max\limits_{X_k \in pa(X_j)} (level(X_k)) + 1 & : \text{otherwise}. \end{cases}$$

The sequential search is conducted in the order of the levels from low to high. This way, the level of htSNPs in T is never greater than that of the currently examined node. As mentioned before, there are two advantages to this ordering: the value of child nodes depends primarily on the information of their parents, and when parents are htSNPs, the child's posterior probability is obtained directly from the network's parameters.

The second heuristic is for expediting the identification of predictive htSNPs for each tagged SNP. That is, if the current set of htSNPs, T, shows a prediction accuracy greater than a pre-specified threshold, β, for SNP X_j, we do not re-evaluate it any more. We formally define the current htSNP set T as the *prediction blanket* of X_j, and use it as the final set of predictive htSNPs for X_j. This second heuristic stems from an empirical observation that when the prediction accuracy for tagged SNP, X_j, given the current set T, is sufficiently high, new htSNPs often do not significantly improve the accuracy. This phenomenon was also observed by others (Ackerman *et al*., 2003). Thus, it is typically unnecessary to examine the effect of every new htSNP on the tagged SNPs that are already well-predicted. The loss in accuracy is typically negligible. Moreover, the potential overfitting of predictive htSNP selection to the training data D is also reduced. Formally, we define the *prediction blanket* as follows:

DEFINITION 4. *Given a prediction indicator function, P_f, and a constant β, the current set of htSNPs, $T = \{X_{t_1}, \ldots, X_{t_q}\}$, is defined as the prediction blanket of X_j if the average prediction accuracy for X_j, over all haplotypes D_{i-} given T is greater than β, that is*:

$$\left[\frac{1}{n}\sum_{i=1}^{n} P_f(X_j, T, D_{i-})\right] > \beta.$$

As a matter of fact, in a Bayesian network, re-evaluation can be avoided whenever T_{X_j} is the Markov blanket of X_j, as information about newly-added htSNPs does not affect the posterior probability of X_j given its Markov blanket. However, it is unlikely that *all* parents, *all* children, and *all* spouses of X_j (i.e., the complete Markov Blanket of X_j) will be included in the current htSNP set T, unless T is very large. Thus, our prediction blanket can be viewed as a relaxed version of the Markov blanket in the context of prediction. The selection algorithm is summarized in Table 1.

4.3 Reconstruction of newly-genotyped samples

The ultimate purpose of prediction-based htSNP selection is to reconstruct the information for all SNPs on a haplotype, using only the selected htSNPs in newly-genotyped samples (for instance, in new association studies). We propose a practical framework for this reconstruction. Our reconstruction algorithm takes *genotype* data of htSNPs as input, infers their resolving haplotypes[5] based on the previously used haplotype data set D, predicts

[5]As defined in the first paragraph of Section 2.

Table 1. BNTagger: Haplotype tagging SNP selection algorithm

D: training data (n haplotypes with p SNPs)
P_f: a prediction indicator function
V: a set of p SNPs $\{X_1, X_2, \ldots, X_p\}$
T: a set of htSNPs $\{T_{t_1}, \ldots, T_{t_q}\}$

// predefined constants
α: accuracy threshold for htSNPs
β: accuracy threshold for prediction blanket

level[X_j]: the *level* of X_j in the BN
status[X_j]: the status of X_j
accuracy[X_j]: the prediction accuracy for X_j

```
Function SequentialSearch (D, P_f){/* Main function */
  T = φ;
  ∀_j status[X_j] = 'unchecked';
  ∀_j accuracy[X_j] = 0;

  L = max_j level[X_j];
  for (each level 1 ≤ l ≤ L)
    for (each node X_j whose level is l)
      accuracy = (1/n) Σ_{i=1}^{n} P_f(X_j, T, D_{i-});
      if (accuracy < α)
        // add this node as an htSNP
        status[X_j] = 'htSNP';
        T = T ∪ {X_j};
        call RevisingSearch(level[X_j]);
      else if (accuracy > β)
        // the prediction blanket of X_j is found
        status[X_j] = 'blanket_found';
        prediction_blanket[X_j] = T;
      else
        // store a candidate predictive htSNPs
        status[X_j] = 'tagged';
        prediction_blanket[X_j] = T;
        accuracy[X_j] = accuracy;
  }

  Function RevisingSearch (L) {
    for (each node X_k
            whose level ≤ L and status = 'tagged')
      accuracy = (1/n) Σ_{i=1}^{n} P_f(X_k, T, D_{i-});
      if(accuracy > β)
        status[X_j] = 'blanket_found';
        prediction_blanket[X_k] = T;
      else if (accuracy > accuracy[X_k])
        prediction_blanket[X_k] = T;
        accuracy[X_k] = accuracy;
  }
```

the alleles of tagged SNPs using the Bayesian network model built in stage I, and outputs the *haplotype* information of *all* SNPs.

Suppose that our htSNP set T, as identified in stage II, consists of q SNPs, that is, $T = \{X_{t_1}, \ldots, X_{t_q}\}$. Let $g = (x_{t_{11}}/x_{t_{12}}, \ldots, x_{t_{q1}}/x_{t_{q2}})$ be a new *genotype*, consisting of the combined allele information of the q htSNPs. To deduce the haplotype information of g, we first select the most common haplotype in D, whose htSNP information is *compatible* with g. The *complementary mate* of the haplotype can then be automatically constructed. If we cannot find any haplotype compatible with g in D, we create a new haplotype whose alleles are assigned as the major allele for each heterozygous htSNP. Let h'_n be the new haplotype, and h'_{n_i} be its i^{th} element (where $i = 1, \ldots, q$). Given $g = (x_{t_{11}}/x_{t_{12}}, \ldots, x_{t_{q_1}}/x_{t_{q_2}})$ h_{n_i} can then be defined as:

$$h'_{n_i} = \begin{cases} x_{t_{i1}} & : \textit{if } x_{t_{i1}} = x_{t_{i2}}; \\ \underset{x \in \{x_{t_{i1}}, x_{t_{i2}}\}}{argmax} Pr(X_{t_i} = x) & : \textit{otherwise}. \end{cases}$$

The prior probability, $Pr(X_{t_i})$, can be computed using our Bayesian network model. Again, its complementary mate can then be automatically constructed. In either case, the inferred two haplotypes for g are separately used for predicting the alleles of each tagged SNP. We call this procedure *incremental* haplotype reconstruction.

The principle of incremental haplotype reconstruction is based on Clark's parsimony approach (Clark, 1990). That is, it tries to resolve an ambiguous genotype using one of the *already identified* haplotypes. Moreover, rather than picking any compatible haplotype, it selects the most common one, since common haplotypes are the most likely candidates under the random mating assumption. Our haplotype reconstruction for the htSNP genotype thus follows the widely-used maximum parsimony approach. However, it differs from conventional algorithms in utilizing the *existing* haplotype information of *all* previously known SNPs, rather than directly phasing those in the genotype. We believe that utilizing this *prior* haplotype information is necessary. As noted earlier, haplotype phasing based on the set of htSNPs might not be as reliable as haplotype phasing based on the original set of SNPs due to the reduced linkage disequilibrium among htSNPs (Halperin *et al*., 2005).

Once the haplotype information of htSNPs is deduced, we use the same prediction rule introduced in Section 2 to predict the tagged SNPs. That is, the allele whose conditional probability is the highest given the alleles of the htSNPs is taken to be the allele for each tagged SNP. When multiple solutions exist, the most common allele of the tagged SNP is selected.

5 RESULTS

5.1 Evaluation methods

We compare the performance of our method with that of three state-of-the-art htSNP selection methods: 1) the Eigen2htSNP method based on principal component analysis (PCA) (Lin and Altman, 2004); 2) the Block-free method based on dynamic programming (Bafna *et al*., 2003; Halldórsson *et al*., 2004); and 3) the STAMPA method based on dynamic programming (Halperin *et al*., 2005). Lin and Altman (2004) tested Eigen2htSNP with two options: *varimax* and *greedy*, and predicted each tagged SNP using the *one* htSNP whose correlation coefficient with the tagged one is the highest. Bafna *et al*. (2003) and Halldórsson *et al*. (2004) tested the Block-free method with two window sizes: 21 and 13, and used the majority vote of htSNPs to predict each tagged SNP. Halperin *et al*. (2005) also relied on the majority vote of htSNPs for prediction, but unlike the previous two methods, they used the *genotype* data of htSNPs rather than haplotype data.

All these methods aim to select a set of highly predictive htSNPs for the unselected, tagged SNPs. Therefore, they have all been evaluated using prediction accuracy. Accordingly, this is the measure we use here for a fair comparison. We note that the published results (Bafna *et al*., 2003; Halldórsson *et al*., 2004; Lin and Altman, 2004; Halperin *et al*., 2005) were all based on different data sets. To compare BNTagger with each of these methods, we obtained the data set used to test each method, preprocessed it as described in the respective publication, and applied our algorithm to it. For evaluation, we use the same evaluation procedure used by each of the compared methods utilizing *leave-one-out* for the Block-free and the STAMPA methods (Bafna *et al*., 2003; Halldórsson *et al*., 2004; Halperin *et al*., 2005) and 10-*fold* cross

Table 2. Summary of test data sets

Data	Data Source	SNP No	Haplotype No	Phasing	Gene Diversity	LD (Std)	Recombination
ACE	Lin and Altman (2004)	52	22	PHASE	0.876	0.78 (0.34)	19.38%
LPL	Nickerson *et al*. (2000)	87	142	known	0.991	0.55 (0.35)	55.95%
IBD5-1	Lin and Altman (2004)	103	774	PHASE	0.981	0.53 (0.27)	94.3%
IBD5-2	Daly *et al*. (2001)	103	258	GERBIL	0.724	0.41 (0.23)	99.6%

validation for Eigen2htSNP (Lin and Altman, 2004), as described in the respective publications. As Lin and Altman (2004) did not provide their 10-fold split, we ran the 10-fold cross validation procedure 10 times, each using a randomized 10-way split, to ensure robustness. In all cases, the average prediction accuracy is used as the ultimate evaluation measure. The prediction performance of the compared methods for each data set was directly taken from their respective publications (Bafna *et al*., 2003; Halldörsson *et al*., 2004; Lin and Altman, 2004; Halperin *et al*., 2005).

5.2 Test data

Three public data sets, ACE (angiotensin converting enzyme) (Rieder *et al*., 1999; Lin and Altman, 2004), LPL (human lipoprotein lipase) (Nickerson *et al*., 2000; Bafna *et al*., 2003; Halldörsson *et al*., 2004), and IBD5 (inflammatory bowel disease 5) (Daly *et al*., 2001; Lin and Altman, 2004; Halperin *et al*., 2005) were used for evaluation. These data sets were previously used to test the three compared methods, as reported in their respective publications. We first analyzed the genetic characteristics of each data set based on: gene diversity, linkage disequilibrium, and recombination rate. The gene diversity, (i.e., the probability that two haplotypes chosen at random from the sample are different (Nei, 1987)), is measured by $(n/(n-1)) \cdot (1 - \sum_{i=1}^{k} p_i^2)$, where n is the total number of haplotypes, k is the number of distinct haplotypes, and p_i is the relative frequency of the i^{th} distinct haplotype. Linkage disequilibrium (LD) between SNPs is estimated by the multi-allelic extension of Lewontin's LD, D' as defined earlier (Hedrick, 1987), where the statistical significance of the standardized LD parameter is calculated using the χ^2 test with one degree of freedom. The recombination rate of each data set is measured by the four-gamete test (Hudson and Kaplan, 1985).

The first data set ACE (Rieder *et al*., 1999) contains 78 SNPs within a genomic region of 24*Kb* on chromosome 17*q*23. Genotyping was done from 11 individuals. This data set was used by Lin and Altman to test Eigen2htSNP (Lin and Altman, 2004). Following their procedure, among the 78 original SNPs only 52 bi-allelic nonsingletons are analyzed. Partially due to the small number of SNPs and small sample size, this data set shows high average LD (0.78) and relatively low gene diversity (0.876). The recombination rate is also relatively low (19.38%).

The second data set LPL (Nickerson *et al*., 2000), which was used by Bafna *et al*. (2003) and Halldörsson *et al*. (2004) to test the Block-free method, contains 88 SNPs spanning 5.5*Kb* on chromosome 19*q*13.22. Genotyping was performed over 71 individuals. Following the analysis performed by Bafna *et al*. (2003), we analyze only 87 bi-allelic SNPs. Despite the small size of the LPL gene, this data set has high gene diversity (0.99) and low average LD (0.55), because it consists of haplotypes from three different populations. The four-gamete test shows 55.95% recombination or recurrent mutation.

The third data set, IBD5 (Daly *et al*., 2001) contains 103 SNPs on chromosome 5*q*31, spanning 500*Kb*. Genotyping was performed over 129 father-mother-child trios from a European population. This data set was used by Halperin *et al*. and by Lin and Altman to test the STAMPA (Halperin *et al*., 2005) and the Eigen2htSNP (Lin and Altman, 2004) methods, respectively. Lin and Altman (2004) analyzed data from all 387 individuals using PHASE (Stephens *et al*., 2001) for haplotype phasing. Halperin *et al*. (2005) analyzed data of only 129 individuals using GERBIL (Kimmel and Shamir, 2005) for haplotype phasing. Thus, following both of these two procedures, we created two separate data sets from IBD5, denoted as IBD5-1 (for Lin and Altman's) and IBD5-2 (for Halperin's). Both these sets have low linkage disequilibrium and high recombination rates. The summary of all data sets is given in Table 2.

5.3 Test results

We summarize the performance of BNTagger compared with the three state-of-the-art htSNP selection methods in Figure 3. We also compute the p-value of the difference in performance, using the Wilcoxon-ranksum test with 5% significance level. Overall, BNTagger consistently outperforms other methods on all data sets. Most importantly, improvement in prediction performance is most notable when the number of selected htSNPs is small, the average linkage disequilibrium in a data set is relatively low, and the gene diversity is high. This is a major advantage of BNTagger, since most htSNP selection methods have been known to suffer in those cases (Crawford and Nickerson, 2005; Johnson *et al*., 2001; Avi-Itzhak *et al*., 2003; Ao *et al*., 2005; Carlson *et al*., 2004). In other words, BNTagger retains its good performance even in what are considered to be hard cases.

The prediction performance of Eigen2htSNP (Lin and Altman, 2004) is compared with ours using two data sets: ACE and IBD5-1. For the first data set, ACE, Eigen2htSNP-varimax shows performance comparable to ours (see Figure 3(a); p-values are 0.2933 for varimax and 4.88×10^{-2} for greedy), but in the case of IBD5-1, its performance is considerably lower than ours, as shown in Figure 3(c) (p-values are 1.9489×10^{-6} for varimax and 1.5707×10^{-8} for greedy). The prediction performance of the Block-free method (Bafna *et al*., 2003; Halldörsson *et al*., 2004) is compared with ours using the LPL data set. Their performance increases substantially with the number of selected htSNPs, as shown in Figure 3(b), but the performance difference between ours and the Block-free method is significant when the number of htSNPs is smaller than 30 (p-values are 4.2×10^{-3} for window 21 and 1.2552×10^{-9} for window 13). The prediction performance of STAMPA (Halperin *et al*., 2005) is compared

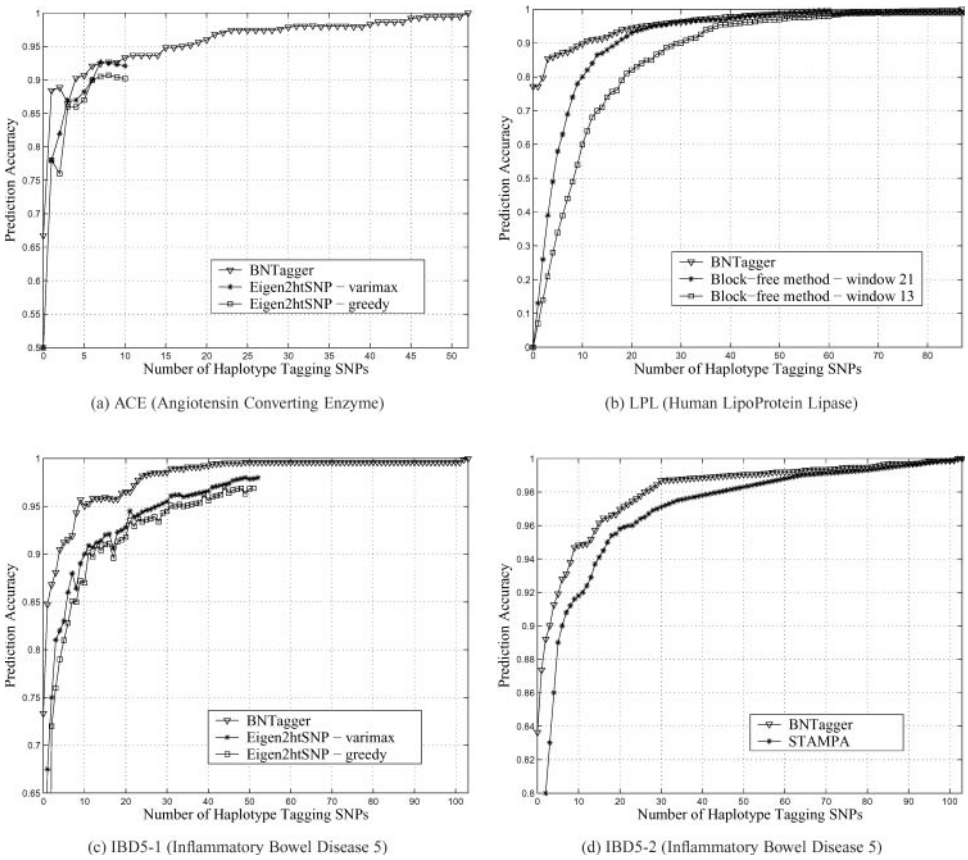


Fig. 3. Prediction performance of BNTagger and the compared methods for test data sets.

Table 3. Prediction accuracy (in %) of BNTagger

Data Set	Percentage of Selected htSNPs				
	0%	5%	10%	25%	50%
ACE	66.7	86.5	92.1	93.7	97.4
LPL	77.2	86.6	89.0	95.0	98.3
IBD5-1	73.3	91.2	95.3	98.4	99.6
IBD5-2	83.6	91.9	94.9	98.0	99.0

with ours using the data set that Halperin *et al*. used, IBD5-2, as shown in Figure 3(d). Again, BNTagger outperforms STAMPA (p-value $= 0.7 \times 10^{-2}$), and the difference is significant as the number of htSNPs gets smaller (below 60).

Overall, as shown in Figure 3, our method uses a small fraction of SNPs as htSNPs (2.9%–11.5%) to achieve 90% prediction accuracy for all data sets: 4 htSNPs among 52 SNPs (7.7%) for data set ACE, 10 among 87 (11.5%) for LPL, 4 among 103 (3.9%) for IBD5-1, and 3 among 103 (2.9%) for IBD5-2. To achieve 95% prediction accuracy, we need 8.7%–32.7% of the target SNPs: 17 htSNPs among 52 SNPs (32.7%) for data set ACE, 22 among 87 (25.2%) for LPL, 9 among 103 (8.7%) for IBD5-1, and 13 among 103 (12.6%) for data set IBD5-2. Table 3 summarizes the prediction performance of BNTagger with respect to the percentage of the selected htSNPs.

As can be seen in Table 3, BNTagger can be reliably used even when the maximum number of htSNPs is very small. This is a major advantage of BNTagger. The explicit goal of htSNP selection is to save genotyping overhead, typically aiming at a 10–50 *fold* reduction in the number of target SNPs in the case of European samples (Palmer and Cardon, 2005). Thus, it is especially important to guarantee good prediction performance when the number of htSNPs is a small fraction of the total number of SNPs. We note that, unlike other methods, BNTagger can predict the allele information of all SNPs even without any htSNPs. In this case, the posterior probability of the predicted SNP X_j is the same as the prior probability of X_j. Thus, the prediction used by the function P_f, as shown in Definition 1, is still applicable even without selecting any htSNPs.

6 DISCUSSION

We presented BNTagger, a heuristic algorithm that uses the probabilistic framework of Bayesian networks to effectively identify a set of predictive htSNPs. BNTagger outperforms other state-of-the-art predictive methods when compared over their own data sets and prediction measure. Moreover, its improved performance is especially notable when a small number of htSNPs are selected. We believe that two main factors contribute to this improved performance:

(1) We do not restrict the htSNPs to any bounded location.

(2) We do not fix the number of htSNPs.

In addition, heuristics based on the conditional independencies among SNPs guide BNTagger to effectively find an improved set of htSNPs in terms of prediction accuracy.

Another major advantage of BNTagger is that, after the htSNPs are selected, it can directly reconstruct the *haplotype* information of newly-*genotyped* samples. BNTagger does not require prior haplotype phasing of htSNPs, which might not be reliable (Halperin *et al*., 2005). Instead, it deduces the haplotype information of the new sample based on the haplotype training data that was originally used for htSNP selection. In addition, BNTagger does not require SNPs to be bi-allelic nor does it assume prior block-partitioning. Nevertheless, it shows significant improvement in prediction performance for data sets with high gene diversity and relatively low linkage disequilibrium. Thus, we believe that BNTagger provides the most practical and comprehensive framework for htSNP selection, and can form a reliable basis for subsequent disease-gene association studies.

The improved performance of BNTagger comes at the cost of compromised running time. Currently, its running time varies from several minutes (when the number of SNPs is 52) to 2–4 hours (when the number is 103). Most of this time is spent on stage I, namely, learning the Bayesian network, rather than on htSNP selection or on haplotype reconstruction. As BNTagger does not partition the haplotype data (neither through blocks nor through a sliding-window[6]), it considers all SNPs at once. That is, the conditional independence structure among all SNPs is learned simultaneously, which substantially increases its running time as the number of SNPs increases. In practice, we argue that based on the clinical importance of disease-gene association studies (Crawford and Nickerson, 2005), improved prediction performance takes priority over running time—when the time is not prohibitively long. Nevertheless, our future research will focus on improving the speed of BNTagger, while minimizing loss in prediction performance. This will most likely involve the evaluation of alternative heuristics and optimization criteria. We also plan to provide BNTagger as an online service.

Currently, BNTagger does not directly set the number of selected htSNPs. Rather, it selects htSNPs based on their prediction accuracy compared to a predefined threshold (α). Thus, by adjusting this threshold, the number of selected htSNPs can be changed. We intend to revise our selection algorithm so that the number of htSNPs can be explicitly set, if needed. Finally, we used the multi-allelic extension of Lewontin's linkage disequilibrium (LD), D' (Hedrick, 1987), to expedite the learning procedure in stage I. We plan to apply other multi-allelic LD measures, and examine whether different measures affect the learned networks, the selected set of htSNPs, and their prediction performance.

ACKNOWLEDGEMENT

This work is supported by HS's NSERC Discovery grant 298292-04 and CFI New Opportunities Award 10437.

[6]Sliding-window-based algorithms confine the predictive htSNPs for each tagged SNP to the ones in the pre-defined neighborhood (i.e., sliding-window) of the tagged SNP (Meng *et al*., 2003).

REFERENCES

Ackerman,H. *et al*. (2003) Haplotype analysis of the TNF locus by association efficiency and entropy. *Genome Biol*., **4**, R24.1–13.

Ao,S.I. *et al*. (2005) CLUSTAG: hierarchical clustering and graph methods for selecting tag SNPs. *Bioinformatics*, **21**, 1735–1736.

Aulchenko,Y. *et al*. (2003) miLD and booLD programs for calculation and analysis of corrected linkage disequilibrium. *Ann Hum Genet*., **67**, 372–375.

Avi-Itzhak,H.I., Su,X. and De La Vega,F.M. (2003) Selection of minimum subsets of single nucleotide polymorphisms to capture haplotype block diversity. *In Proc. of Pac Symp Biocomput*., 466–477.

Bafna,V. *et al*. (2003) Haplotypes and informative SNP selection algorithms: don't block out information. *In Proc. of Intl Conf Res Comp Mol Biol*., 19–27.

Carlson,C.S. *et al*. (2004) Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium. *Am J Human Genet*., **74**, 106–120.

Clark,A.G. (1990) Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol Biol Evo*., **7**, 111–122.

Cozman,F. (2000) Generalizing variable elimination in Bayesian networks. *In Proc. of the Workshop on Probabilistic Reasoning in Artificial Intelligence*, 27–32.

Crawford,D. and Nickerson,D. (2005) Definition and clinical importance of haplotypes. *Annu Rev Med*., **56**, 303–320.

Daly,M. *et al*. (2001) High-resolution haplotype structure in the human genome. *Nat Genet*., **29**, 229–232.

De Bakker,P.I.W. *et al*. (2006) Transferability of tag SNPs to capture common genetic variation in DNA repair genes across multiple populations. *In Proc. of Pac Symp Biocomput*., 478–486.

Friedman,N., Nachman,I. and Peér,D. (1999) Learning bayesian network structure from massive datasets: the ''sparse candidate'' algorithm. In *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence (UAI),* 206–215.

Gabriel,S. *et al*. (2002) The structure of haplotype blocks in the human genome. *Science*, **296**, 2225–2229.

Greenspan,G. and Geiger,D. (2003) Model-based inference of haplotype block variation. *In Proc. of Intl Conf Res Comp Mol Biol*., 131–137.

Halldórsson,B.V. *et al*. (2004) Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies. *Genome Res*., **14**, 1633–1640.

Halldórsson,B.V. *et al*. (2004b) A survey of computational methods for determining haplotyes. *Lecture Notes in Computer Science* **2983**, 26–47.

Halperin,E., Kimmel,G. and Shamir,R. (2005) Tag SNP selection in genotype data for maximizing SNP prediction accuracy. *Bioinformatics*, **21** (Suppl. 1), i195–i203.

Hedrick,P. (1987) Gametic disequilibrium measures: proceed with caution. *Genetics*, **117**, 331–341.

Hudson,R. and Kaplan,N. (1985) Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, **111**, 147–164.

Jensen,F. (2002) *Bayesian networks and decision graphs*. In M. Jordan, S.L. Lauritzen, J.F. Lawless and V. Nair (eds), Springer-Verlag, New York.

Johnson,G.C.L. *et al*. (2001) Haplotype tagging for the identification of common disease genes. *Nat Genet*., **29**, 233–237.

Kimmel,G. and Shamir,R. (2005) GERBIL: genotype resolution and block identification using likelihood. *Proc. Natl Acad Sci*., **102**, 158–162.

Lam,W. and Bacchus,F. (1994) Learning bayesian belief networks: an approach based on the MDL principle. *Comp Intel*., **10**, 269–293.

Lin,Z. and Altman,R.B. (2004) Finding haplotype tagging SNPs by use of principal components analysis. *Am J Human Genet*., **75**, 850–861.

Meng,Z. *et al*. (2003) Selection of genetic markers for association analyses using linkage disequilibrium and haplotypes. *Am J Human Genet*., **73**, 115–130.

Nei,M. (1987) *Molecular evolutionary genetics*. Columbia University Press, New York.

Nickerson,D. *et al*. (2000) Sequence Diversity and Large-Scale Typing of SNPs in the Human Apolipoprotein E Gene. *Genome Res*., **10**, 1532–1545.

Palmer,L. and Cardon,L. (2005) Shaking the tree: mapping complex disease genes with linkage disequilibrium. *Lancet*, **366**, 1223–1234.

Reich,D. *et al*. (2001) Linkage disequilibrium in the human genome. *Nature*, **411**, 199–204.

Rieder,M. *et al*. (1999) Sequence variation in the human angiotensin converting enzyme. *Nat Genet*., **22**, 59–62.

Stephens,M., Smith,N. and Donnelly,P. (2001) A new statistical method for haplotype reconstruction from population data. *Am J Human Genet*., **68**, 978–989.

Xing,E.P., Sharan,R. and Jordan,M.I. (2004) Bayesian haplotype inference via the Dirichlet process. In *Proc. of the 21st International Conference on Machine Learning,* 879–886.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e220–e226
doi:10.1093/bioinformatics/btl203

Finding the evidence for protein-protein interactions from PubMed abstracts

Hyunchul Jang[1,*], Jaesoo Lim[1], Joon-Ho Lim[1], Soo-Jun Park[1], Kyu-Chul Lee[2] and Seon-Hee Park[1]

[1]Bioinformatics Team, Electronics and Telecommunications Research Institute (ETRI), Gajeong-Dong, Yuseong-Gu, Daejeon, 305-350, Korea and [2]Department of Computer Engineering, Chungnam National University, Gung-Dong, Yuseong-Gu, Daejeon, 305-764, Korea

ABSTRACT

Motivation: Protein-protein interactions play critical roles in biological processes, and many biologists try to find or to predict crucial information concerning these interactions. Before verifying interactions in biological laboratory work, validating them from previous research is necessary. Although many efforts have been made to create databases that store verified information in a structured form, much interaction information still remains as unstructured text. As the amount of new publications has increased rapidly, a large amount of research has sought to extract interactions from the text automatically. However, there remain various difficulties associated with the process of applying automatically generated results into manually annotated databases. For interactions that are not found in manually stored databases, researchers attempt to search for abstracts or full papers.

Results: As a result of a search for two proteins, PubMed frequently returns hundreds of abstracts. In this paper, a method is introduced that validates protein-protein interactions from PubMed abstracts. A query is generated from two given proteins automatically and abstracts are then collected from PubMed. Following this, target proteins and their synonyms are recognized and their interaction information is extracted from the collection. It was found that 67.37% of the interactions from DIP-PPI corpus were found from the PubMed abstracts and 87.37% of interactions were found from the given full texts.

Availability: Contact authors.

Contact: janghc@etri.re.kr

1 INTRODUCTION

An uncountable number of protein-protein interactions are buried in research papers published thus far, and the number of papers published is growing continuously. Although there are stored data in verified databases such as BIND (Bader *et al*., 2001), KEGG (Kanehisa *et al*., 2002), SwissProt (Bairoch *et al*., 2000), and the Database of Interacting Proteins (Xenarios *et al*., 2001), these sources occasionally do not satisfy researchers. Even if the data is very useful, easily searchable and well structured, these databases nonetheless do not store the whole data, and most of protein interactions remain as unstructured text from scientific abstracts and full papers (Blaschke *et al*., 2001, 2002; Temkin *et al*., 2003). Moreover, most of the data exist only in the scientific literature. They are scattered in throughout the scientific literature and written in natural language. Accordingly, automated extraction information from the PubMed abstracts is preferable, and research that consolidates the set of known protein interactions using biomedical literature is necessary (Jenssen *et al*., 2001; Hirschman *et al*., 2002; Rzhetsky *et al*., 2004; Ramani *et al*., 2005).

In recent years, many researches have proposed to extract the information regarding protein interactions with automatic tools. However key issues such as the detection of protein names are not completely resolved with the use of such tools, thus they remain far from perfect (Blaschke *et al*., 2001, 2002).

Various techniques for recognizing protein names have been proposed. The use of standardized dictionaries containing the names and synonyms of proteins has been shown to be effective for recognizing these entities in text (Blaschke *et al*., 1999; Rindflesch *et al*., 1999, 2000). This technique remains limited as protein names not present in the dictionaries produce large amounts of false negatives. Others have proposed approaches using templates capable of recognizing common naming patterns for proteins (Fukuda *et al*., 1998; Ng *et al*., 1999; Yu *et al*., 2002). These techniques have also been shown to generate a large number of false positives by recognizing words that match the templates but are in fact not proteins. Alternative approaches have proposed machine learning methods (Proux *et al*., 1998; Hatzivassiloglou *et al*., 2001), and statistical methods (Krauthammer *et al*., 2000; Tanabe *et al*.,, 2002). Although these techniques have reported incremental gains in overall recall and precision over the template and dictionary based approaches, it has been shown that these techniques are also limited by the quality and extent of the training sets used to train the algorithms (Tanabe *et al*., 2002).

Similar to the limits inherent in the recognition of protein names, there have been various approaches published for extracting relationships from scientific literature. Several researches have shown that template and simple rule based algorithms can be used to extract interactions (Sekimizu *et al*., 1998; Blaschke *et al*., 1999; Ng and Wong 1999; Thomas *et al*., 2000; Friedman *et al*., 2001; Ono *et al*., 2001; Wong 2001; Pustejovsky *et al*., 2002). These approaches are, however, limited to a set of interactions by the predefined extraction rules or templates. Complicated cases are often

*To whom correspondence should be addressed.

missed by these approaches. Others have proposed the use of parts of speech analysis (Humphreys *et al*., 2000), and natural language based approaches (Rindflesch *et al*., 2000; Friedman *et al*., 2001). Huang *et al*., proposed a method for automatically generating patterns and extracting protein interactions (Huang *et al*., 2004; Hao *et al*., 2005). Bunescu *et al*., showed that various rule induction methods are able to identify protein interactions with higher precision than manually-developed rules (Bunescu *et al*., 2004). Ramani *et al*., used a set of 230 Medline abstracts manually tagged for both proteins and interactions to train an interaction extractor (Ramani *et al*., 2005). However, machine learning techniques are also limited by the quality and extent of the training sets used to train the algorithms.

A lack of standard common corpus, techniques and equations for reporting recall and precision has made comparative analysis of different approaches a difficult job (Hirschman *et al*., 2002).

Most of the current biological knowledge can be retrieved from the MEDLINE database, which now has records from more than 4,800 journals accounting for nearly 15 million articles. These citations contain thousands of experimentally recorded protein interactions. However, because of the large number of articles and the lack of formal structure, it is difficult to retrieve the data. A method to validate given protein-protein interactions from PubMed abstracts with the limits listed above is proposed.

2 METHODS

The present protein-protein interaction validation system consists of the following components, as shown in Fig. 1:

(i) A PubMed collector

(ii) A PPI extractor

(iii) A PPI validator

The abstracts collection component generates a PubMed query from the given two protein names and then collects abstracts from PubMed. The interaction extraction phase divides abstracts into sentences and recognizes protein names in sentences. Following this, sentences that have both proteins are selected, morphologically tagged and syntactically parsed after sentence simplification. As the last step of the extraction component, interactions between two proteins are extracted from the syntactically parsed sentences. The conflict resolution component detects false-positive interactions that were extracted, removes these false interactions, and decides whether the wanted interaction exists.

Brill's transformation-based part-of-speech tagger[1] (Brill 2002) was utilized, and was trained with the GENIA corpus[2] (Kim *et al*., 2003). Its precision was 98.35% after training with the GENIA corpus and 83.73% with the WSJ corpus. The Stanford Parser[3] version 1.4 with probabilistic context free grammar (PCFG) was also used.

2.1 PubMed abstracts collection

Simple queries for two proteins were generated in the forms of ''A and B''. In addition, two proteins A and B are expanded automatically with their synonyms. In the query step, users can add additional missed synonyms or abbreviations. The final query strings are in a form that resembles ''(A or A1 or A2 or . . . or Aa) and (B or B1 or B2 or . . . or Bb)'' under 'A1', 'A2', . . .

[1]Eric Brill's Home Page: http://www.cs.jhu.edu/~brill/RBT1_14.tar.Z

[2]GENIA corpus: http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/

[3]The Stanford Natural Language Processing Group: http://www-nlp.stanford.edu/software/lex-parser.shtml

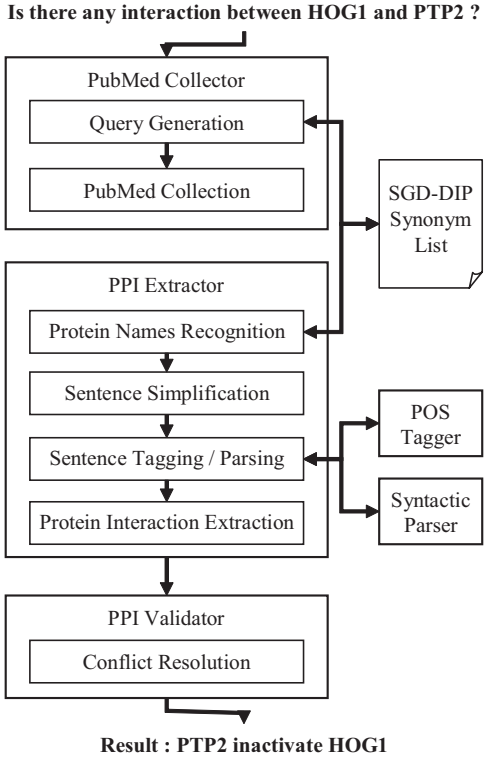


Fig. 1. System overview.

'Aa' and 'B1', 'B2', . . . 'Bb' are the synonyms of protein A and B. Fig. 2 shows a flowchart for the abstract collection phase.

The proposed system searches PubMed through the use of Entrez Utilities[4] and collects PubMed abstracts with parsed ID lists from the results under the site's user requirements. If the number of searched abstracts is small, a user may regenerate the query or may read the abstract directly. The PubMed collector stores titles and abstract texts from the abstracts fetched in XML from PubMed.

2.2 PPI extraction

The sentences are parsed syntactically and interactions are extracted from them. The result of a parser in the form of the Penn Treebank syntactic tags (Marcus *et al*., 1994) is then applied. Fig. 3 (a) is an example sentence, and Fig. 3 (b) shows the parsing result for it. This shows the syntactic tree structure and how the interaction is extracted between two proteins through the traversing of the tree. This is similar to finding a path between two leaf nodes.

Many existing full parsers that are not tuned to the biomedical domain frequently fail to parse, or their parsed results are often incorrect. This result occurs as most sentences in the biomedical literature are syntactically complex, or because words in sentences are tagged incorrectly. The sentence in Fig. 4 (a) is an example of this. This sentence has 43 tokens when the parentheses are tokenized and the minus symbols are not tokenized. To avoid this problem, sentences are made simple by the proposed method by substituting one word for complex words, i.e., protein names and nouns.

Protein names recognition The protein name extractor tags proteins using the words that were used for the PubMed query. Capitalized characters

[4]Entrez Utilities Site: http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html

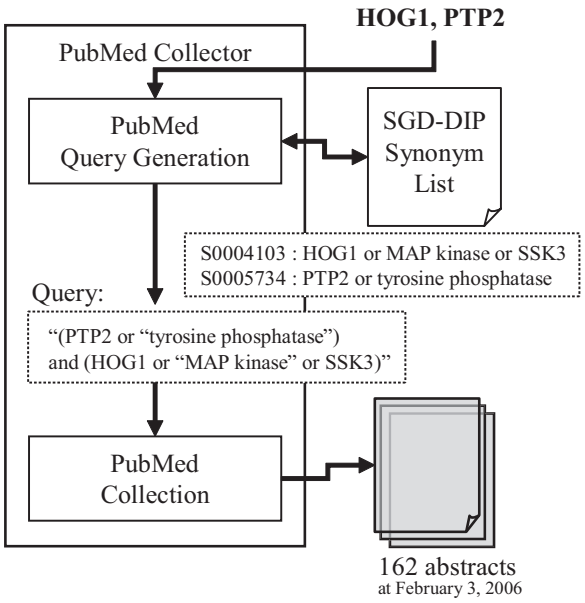


Fig. 2. Abstract collection.

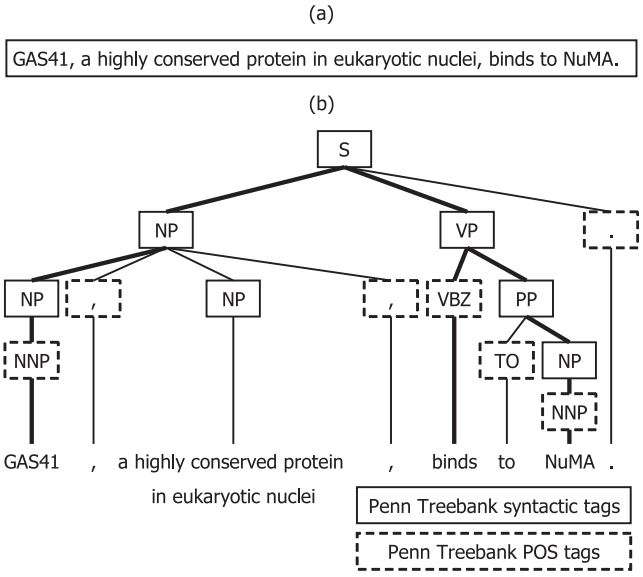


Fig. 3. A sentence and its Penn Treebank syntactic tree.

are ignored in the string matching step. Sentences those have both proteins are selected to extract interactions.

The use of dictionaries containing the names and synonyms of proteins has been shown to be effective for recognizing entities in free form text (Blaschke *et al*., 1999; Rindflesch *et al*., 1999). However, applications of this technique remain limited for the reason that protein names not present in the dictionaries produce large amounts of false negatives. This technique has reported high rates of recall and precision, and the proposed method relates to only two proteins at the validation step.

Making sentences simple Biomedical sentences are generally complex. One reason for this is that named entities in biomedical texts are usually not simple and consist of many words that have various morphological tags. This makes it difficult to parse biomedical texts.

One named entity can be divided into different phrases. This causes the structure to collapse. Accordingly the sentences are made simple by the following steps. First, recognized protein names are substituted with one predefined word. Second, noun phrases are substituted with one predefined word. Third, parenthesis phrases that are not a part of a named entity are removed. Following these steps, a more simplified sentence is created. The sentence in Fig. 4 (a) is changed to that in Fig. 4 (b). The new sentence now has 27 tokens. The parser can then process this sentence correctly. Lexicons were modified to tag substituted named entity words as NNPs and to tag substituted noun phrase words as NNs.

Tagging and parsing sentences Before extracting the protein-protein interactions, sentences are morphologically tagged and syntactically parsed. The tagging results of an in-domain tagger are better than a tagger embedded in a full parser, as a tagger can be trained with a morphologically tagged corpus. However, there is no proper corpus for a full parser in the biomedical domain; therefore, if the parser can receive the results of in-domain tagger, it can produce better parsing results.

The parser returns syntactically tagged sentences as shown in Fig. 4 (c). The tree structure of the sentence in Fig. 4 is shown in Fig. 5. Following this, the proposed extractor can analyze the entire syntactic structure of the sentences. Instead of various templates or patterns from the syntactic tags, the extractor traverses the structured syntactic trees of sentences. The proposed rules can be simple and light, as the syntactic tag set has fewer number of tags than the POS tag set. The structural complexities of sentences are simplified into the tree hierarchies.

Extracting protein-protein interactions It is not straightforward to decide whether extracted paths between leaf nodes of a syntactic tree structure denote meaningful relationships. However, if two leaf nodes are proteins, it is easier to decide whether a relationship between them exists or not. If verbs or nouns are one of the predefined keywords, the extractor considers them as interaction events. The keywords are manually listed and based on the research by Temkin *et al*., and Hakenberg *et al*., (Temkin *et al*., 2003; Hakenberg *et al*., 2005). The keyword list was acquired via the Internet, from the homepage of Jörg Hakenberg[5].

First, the extractor finds NP tags, and then checks whether NP belongs to any of these three cases: NP+VP, NP+PP or NP+CC+NP. Most of the interactions belong to one of these types; others usually belongs the following two cases. The first is similar to 'is-a' semantically, as in: 'JAB has recently been identified as a regulator of JAK2 phosphorylation and activity by binding phosphorylated JAK2 and inducing its degradation.' This sentence contains 'JAB phosphorylates JAK2' information. The second is JJ+NNP, as in: 'CD38-associated Lck'. These two types are processed by a template-based method.

In Fig. 3, the sentence has NP and VP tags. 'GAS41' is the first noun of the first NP in the top NP. The extractor looks initially at the first NP in the top NP and finds the first noun, NNP, in it. It finds the verb, VBZ in the VP and 'binds' is extracted. Finally, it looks for PP after VBZ and finds NP. 'NuMA' is extracted from the NP. This sentence presents in the form of NP+VP. 'GAS41' represents the NP phrase, 'binds' and 'NuMA' represents the VBZ and PP phrases in VP phrase. This is not a passive form and there is no negative expression. Therefore, 'GAS41' is the subject of the 'binds' event and 'NuMA' is the object.

In Fig. 5, a NP+VP structure is detected, and the PPI extractor finds 'NED' as a subject and 'activated' as an event. From the VP that has VBN 'activated', 'NEE' is found as an object. Finally, the subject and object are exchanged due to the IN, 'by' and VBN tags.

Negative expressions can be extracted from the RB or DT tags in any phrase. Each type of required PP phrases was manually defined. Thus, the extractor continues to search for one more PP phrase after extracting 'interaction' and 'FKBP12', shown in Fig. 6. 'RyR1' and 'IP3R1' are found from the next PP phrase.

[5]http://www.informatik.hu-berlin.de/∼hakenber/

(a)

To define the mechanism underlying signal transducer and activator of transcription (STAT) protein recruitment to the interleukin 10 (IL-10) receptor, the STAT proteins activated by IL-10 in different cell populations were first defined using electrophoretic mobility shift assays.

(b)

To define the mechanism underlying NEA NPA to the NEC , the NED proteins activated by NEE in different NPB were first defined using electrophoretic NPC .

(c)

(S (S (VP (TO To) (VP (VB define) (NP (DT the) (NN mechanism)) (NP (NP (VBG underlying) (NNP NEA) (NN NPA)) (PP (TO to) (NP (DT the) (NNP NEC))))))) (, ,) (NP (NP (DT the) (NNP NED) (NNS proteins)) (VP (VBN activated) (PP (IN by) (NP (NP (NNP NEE)) (PP (IN in) (NP (JJ different) (NN NPB))))))) (VP (VBD were) (ADVP (RB first)) (VP (VBN defined) (S (VP (VBG using) (NP (JJ electrophoretic) (NN NPC)))))) (. .))

Fig. 4. (a) Complex sentence, (b) Simplified sentence, and (c) Parsed sentence in Penn Treebank syntactic tag format.

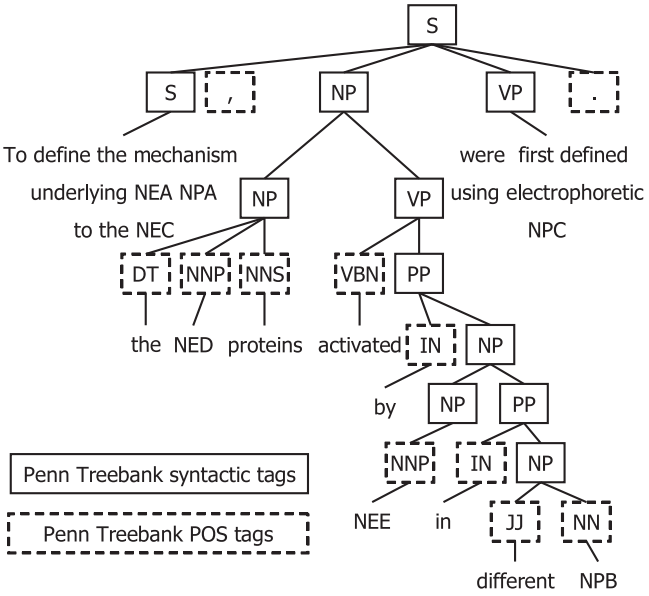


Fig. 5. Penn Treebank syntactic tree.

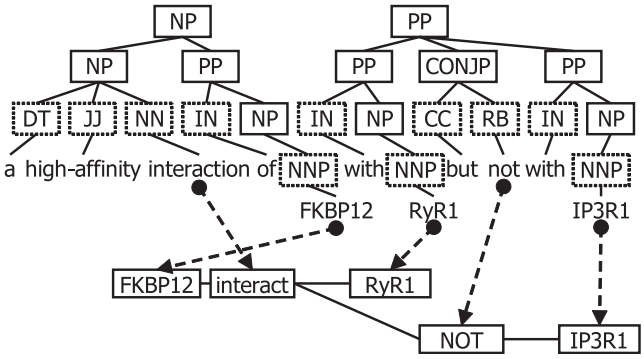


Fig. 6. Complex and negative interactions extraction.

Table 1. Extracted relationships between 'MEK1' and 'ERK2'

Subject string	Event string	Object string	PubMed	Yapex
MEK1	associate	ERK2	2	
MEK1	interact	ERK2	4	1
MEK1	complex	ERK2	2	
MEK1	bind	ERK2		1
MEK1	activate	ERK2	5	1
MEK1	phosphorylate	ERK2	4	1
ERK2	phosphorylate	MEK1	3	1

In all cases, a NNP-tagged protein is extracted as a subject or an object only when it is the first noun in the NP, and this NP is the first NP in its parent NP or PP. If a protein or a NP follows CC, they and their parent are available as a subject or an object.

2.3 Conflict resolution

Protein names recognition and their interactions extraction are very complex, and can be ambiguous. As a result of these processes, false positives in the extracted information may occur. In the proposed method, these are usually caused by parsing errors or rules for high recall.

Protein names conflict One protein name may indicate more than one protein that is different in terms of species. The same string can signify another category. Therefore, it is necessary to confirm that subjects and objects of extracted interactions are the truly wanted proteins.

Conflicts in protein names are caused by false positively recognized names, such as different species or categories, abbreviations, inaccurate boundaries, or homonyms. For now, it is considered that identical strings indicate identical proteins and that there are no distinctions among species and categories.

Relation events conflict For interactions, several types of interactions can be extracted as a false positive, or correctly extracted as two identical proteins, as shown in Table 1. The most critical conflict comes when they are opposites. Incorrectly extracted reverse interactions have to be removed.

Currently detected are only conflicts in which some interactions are positive and others are negative. In addition, interactions by type or polarity are not distinguished.

279 abstracts were collected from PubMed with a query 'MEK1 and ERK2', limited to only items with abstracts. The proposed system extracted 20 interactions between 'MEK1' and 'ERK2' in the abstracts. In the Yapex testing corpus, five interactions were extracted.

Due to the 20 extracted events, the proposed system can validate that interaction between 'MEK1' and 'ERK2' exist. However, understanding whether two phosphorylation interactions, 'MEK1 phosphorylate ERK2' and 'ERK2 phosphorylate MEK1', are in conflict is not easy to determine. In this case, two interactions were correctly extracted when the experimental conditions were ignored. It is nearly impossible to decide that some interactions are not facts.

3 RESULTS AND DISCUSSION

3.1 Full parsing sentences

The Yapex[6] corpus was selected to evaluate the effect of sentence simplification. The Yapex corpus is used for the purpose of evaluating named entity recognition methods. It consists of 99 abstracts for training and 101 abstracts for testing. 101 testing abstracts were utilized for the evaluation. The Yapex testing corpus has 962 sentences, including abstract titles. The number of sentences that have more than two protein names is 532. The parser processed 439 sentences, and did not process 93 sentences. The percentage of parsed sentences was 82.5% and the average number of tokens per sentence was 24.97. The percentage of failed sentences was 17.5% and the average number of tokens per sentence was 49.07.

After sentence simplification, the parser could parse additional 62 sentences, and only 31 of 93 sentences were left out. The average number of tokens was 26.15 in 501 sentences, and 53.38 words in 31 sentences. The parser success rate is higher when the morphological tags are given by the tagger. The precision of parsed results was not evaluated. However, 62 sentences (11.7%) could be parsed after simplification. This indicates that the sentences could be parsed more correctly.

3.2 Extracting protein-protein interactions

The BC-PPI[7] corpus was selected in order to evaluate the proposed protein-protein interaction validation method. This corpus consists of 1,000 sentences, with annotated genes/proteins and interactions. It contains 255 interactions and 173 sentences contain at least one interaction. If a sentence includes more than one interaction, all interactions were counted as answers. Additionally, the present system tried to extract all.

The value of a recall was calculated to be TP/(TP+FN)∗100, and the value of a precision was calculated to be TP/(TP+FP)∗100. TP indicates the total number of interactions extracted correctly and tagged in the corpus, TP+FN indicates the total number of interactions tagged in the corpus, and TP+FP indicates the total number of interactions extracted correctly or incorrectly by the proposed method. The rate of recall and precision of extraction with the sentence simplification were 42.74% and 81.34%, respectively. The BC-PPI corpus has no negatively tagged interaction; hence any extracted negative interactions were excluded from TP+FP. The TP was 109, the TP+FN was 255, and the TP+FP was 134, as shown in Table 3. The proposed method was not evaluated without the sentence simplification. Extracted protein names can be scattered over the syntactic tree and the proposed interaction extraction method does not address this problem.

[6]Yapex corpus: http://www.sics.se/humle/projects/prothalt/

[7]BioCreAtIve-PPI corpus: http://www.informatik.hu-berlin.de/∼hakenber/corpora/

Table 2. Parsing before and after sentence simplification

Sentence simplification	Full parsing Success	Fail
Before	439(82.51%)	93(17.48%)
After NES	455(85.52%)	77(14.47%)
After NES+NPS	474(89.09%)	58(10.90%)
After NES+NPS+PPR	501(94.17%)	31(05.82%)

NES: named entity substitution, NPS: noun phrase substitution, PPR: parenthesis phrase removal.

Table 3. Recall and precision for BC-PPI corpus

TP+FN	TP	TP+FP	Recall	Precision
255	109	134	42.7%	81.3%

Some false positively extracted interactions were caused by parsing fail or error. A parsing failure indicates that the parser can not parse, and parsing error signifies that it does not parse correctly. The false positively extracted interactions are caused by a parsing error, as in: 'We concluded that the two NF-IL6 sites mediate induction of IL-1 beta in response to the stimuli LAN, LPS, and TNF-alpha.' The parser returned 'the two NF-IL6 sites mediate TNF-alpha'.

Most missed interactions are caused by semantic problems. The proposed extractor does not account for semantic relations; as well, and syntactic tags don't indicate them. The following sentences are examples:

(1) ''Receptor activation by the haematopoietic growth factor proteins interleukin 5 (IL-5) and granulocyte-macrophage colony-stimulating factor (GM-CSF) leads to phosphorylation of JAK2 as a key trigger of signal transduction.''

(2) ''We analyzed the abilities of fibrillins and LTBPs to bind latent TGF-beta by their 8-Cys repeats.''

(3) ''In vitro GAS41 bound to the C-terminal part of the rod region of NuMA.''

These sentences need to be handled semantically, or errors occur. For examples, The proposed system was not able to determine that 'leads to phosphorylation of' is equivalent to 'phosphorylate' in sentence (1), or that 'the abilities of fibrillins to bind' corresponds to 'fibrillins binds' in sentence (2). In addition, it did not determine that 'to the C-terminal part of the rod region of NuMA' meant that 'to NuMA' in sentence (3).

Although only a small number of interactions are expressed with anaphora terms, they were not analyzed, though unquestionably this should be addressed. The following sentence is an example of this.

(4) ''Deletion of the binding site from MEK1 reduced its phosphorylation by ERK2, but had no effect on its phosphorylation by p21-activated protein kinase-1 (PAK1).''

Table 4. Number of validated interactions by one abstract, full text, and a number of abstracts

190 interactions	(A)	(B)	(C)
Not Validated	107	24	62
	56.32%	12.63%	32.63%
Validated	83	166	128
	43.68%	87.37%	67.37%

(A) using only one abstract, (B) using full-text and (C) using abstracts collected from PubMed.

3.3 Finding the evidences for PPIs

The DIP-PPI[8] corpus was selected to evaluate the proposed validation method. The DIP-PPI corpus is based on protein-protein interactions from the DIP[9], and is restricted to proteins from yeast. The full texts are included in the corpus, rather than the abstract only. DIP uses IDs from the SGD[10] for nodes. The DIP-PPI corpus contains 297 interactions. For protein synonyms, the DIP synonyms[11] from SGD of the DIP-PPI corpus were used, and a number of missed synonyms and aliases were added from the SGD Gene Names[12].

20 interactions from the DIP-PPI corpus are composed of one protein. These are interactions in which the first partner and the second partner have the same SGD ID, and they were excluded from the validation.

An abstract vs. a full text vs. abstracts In addition, 87 interactions are valid but the corpus contains no text for these. 107 interactions were totally excluded while 190 interactions were included to compare the effects of an abstract, a full text and abstracts for the interactions.

As shown in Table 4, from among 190 interactions, 166 interactions were extracted from the full text given in the corpus, with the rate of 87% as shown in Table 4 (B). When using only each abstract instead of the given full text for an interaction, only 83 interactions were extracted, as shown in Table 4 (A). When using all collected abstracts for an interaction, 128 interactions were extracted, as shown in Table 4 (C). These results show that using a number of collected abstracts for an interaction is more effective naturally compared to using an abstract, and less compared to the use of full text versions.

When abstracts collected from PubMed were used, no abstract was collected for 11 interactions, and no target interaction was extracted from the collected abstracts for 51 interactions. 13 from 51 had no sentence that had both proteins, and 38 from 51 had more than one sentence that had both proteins; however, no wanted interaction was extracted.

PubMed returned at least one abstract for 179 interactions, and abstracts identical to those in the PubMed ID as a given corpus were searched in 128 of 179 interactions. Coincidentally, 128 of 179 interactions were validated; however, this does not indicate that the only interaction in which the same abstract was given in the corpus could be validated.

Table 5. Number of validated interactions from the PubMed abstracts number of abstracts

277 interactions	(D)	(E)	(F)
Not Collected	27		
	9.75%		
No Sentence	29	29	
	10.47%	11.60	
Not validated	57	57	57
	20.58%	22.80%	25.79%
Validated	164	164	164
	59.20%	65.60%	74.21%
Total	277	250	221
	100.00%	100.00%	100.00%

(D) total interactions, (E) interactions that abstracts are collected from PubMed and (F) interactions in which both proteins are found in the sentences

Co-occurrence: found vs. not found No abstract was collected by the query generated in this trial for 27 interactions, and at least one abstract was collected for each of the 250 interactions as shown in Table 5 (D).

In order to validate an interaction between two proteins, the proposed system has to find at least one sentence in which both proteins are present. Among the 250 interactions in Table 5 (E), 221 collections had at least one sentence in which both proteins were present. 164 of 221 interactions that have more than one sentence were validated as shown in Table 5 (F). 57 interactions were not validated from those sentences found in the PubMed abstracts.

In real cases, a user can edit the proposed query for the PubMed collection. However, the query is generated from the given protein names automatically.

In case no relationship is extracted from sentences in which two proteins are present, the co-occurrence information may be useful in a statistical method. However, this was not calculated at this point.

Although more than thirty sentences in which both proteins were present were collected, the interaction between the two proteins could not be validated. Only 11 of 164 interactions were validated from more than thirty sentences. 153 of 164 interactions were validated in less than thirty sentences. This indicates that the validation possibility is not very dependent on the number of collected sentences in which both proteins were present.

From seven invalidated interactions, more than forty abstracts were collected, but the wanted interactions were not extracted. From 131 validated interactions, less than thirty abstracts were collected for each interaction. This signifies that the validation possibility is not overly dependent on the number of abstracts collected.

4 CONCLUSION

A PubMed abstract-based protein-protein interaction validation method is presented. The basic idea of this approach is that sentences in the biomedical literature are simplified after multi-word substitutions. Additionally, a normal full parser can parse these

[8]DIP-PPI corpus: http://www.informatik.hu-berlin.de/~hakenber/corpora/
[9]Database of Interacting Proteins: http://dip.doe-mbi.ucla.edu/
[10]Saccharomyces Genome Database: http://www.yeastgenome.org/
[11]gene/protein names from SGD: http://www.informatik.hu-berlin.de/~hakenber/corpora/dipppi/
[12]http://www.yeastgenome.org/gene_list.shtml

simplified sentences even if the parser is not tuned to biomedical sentences. In the next step, the proposed system reads the results from the parser and extracts all existing interactions. For validation, more than one abstract was used and any extracted interactions that were false positives were resolved.

When the recall performance was assessed through the use of the DIP database of protein–protein interactions, the recall for IntEx and BioRAT were approximately 27% and 20%, respectively (Corney 2004). The recall in this study is 44% when only one abstract is used.

The proposed method validated protein-protein interactions at a rate of 43.68% through the use of one given abstract for an interaction, 67.37% through the use of collected PubMed abstracts, and 87.37% through the use of a given full-text paper. This value is different from the normal recall rate. For collected abstracts with proper sentences, the proposed method validated interactions in nearly 75% of the cases. Additionally, for a case in which at least one abstract was collected, the proposed method validated at a rate of 65%.

ACKNOWLEDGEMENTS

The project described in this paper was fully supported by the Korean Institute for Information Technology Advancement (IITA) under the Korean Ministry of Information and Communication.

REFERENCES

Bader,G.D., Donaldson,I., Wolting,C., Ouellette,B.F., Pawson,T. and Hogue,C.W. (2001) BIND—The Biomolecular Interaction Network Database. Nucleic Acids Res., 29(1), 242–245.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res., 28, 45–48.

Blaschke,C., Andrade,M.A., Ouzounis,C. and Valencia,A. (1999) Automatic extraction of biological information from scientific text: protein-protein interactions. Proceedings of the AAAI Conference on Intelligent Systems for Molecular Biology (ISMB), AAAI Press, 60–67.

Blaschke,C., Oliveros,J.C. and Valencia,A. (2001) Mining functional information associated with expression arrays. Funct Integr Genomics, 1(4), 256–268.

Blaschke,C. and Valencia,A. (2001) Can bibliographic pointers for known biological data be found automatically? Protein interactions as a case study Comp. Funct. Genomics, 2(4), 196–206.

Blaschke,C. and Valencia,A. (2002) The frame-based module of the SUISEKI information extraction system. IEEE Intell. Syst., 17, 14–20.

Brill,E. (2002) Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. Computational Linguistics, 21.4, 543–565.

Bunescu,R., Ge,R., Kate,R.J., Marcotte,E.M., Mooney,R.J., Ramani,A.K. and Wong,Y.W. (2004) Comparative Experiments on Learning Information Extractors for Proteins and the Interactions. Journal of Artificial Intelligence in Medicine, 33, 139–155.

Corney,D.P.A., Buxton,B.F., Langdon,W.B. and Jones,D.T. (2004) BioRAT: extracting biological information from full-length papers. Bioinformatics, 20(17), 3206–3213.

Friedman,C., Kra,P., Yu,H., Krauthammer,M. and Rzhetsky,A. (2001) GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics, 17 (Suppl. 1), S74–S82.

Fukuda,K., Tamura,A., Tsunoda,T. and Takagi,T. (1998) Toward information extraction: identifying protein names from biological papers. Proceedings of the Pacific Symposium on Biocomputing, 98, 707–718.

Hakenberg,J., Plake,C., Lese,U., Kirsch,H. and Rebholz-Schuhmann,D. (2005) LLL'05 Challenge: Genic Interaction Extraction with Alignments and Finite State Automata. Proceedings of Learning Language in Logic Workshop (LLL'05) at ICML, 38–45.

Hao,Y., Zhu,X., Huang,M. and Li,M. (2005) Discovering Patterns to Extract Protein-Protein Interactions from the Literature: Part II. Bioinformatics, 21(15), 3294–3300.

Hatzivassiloglou,V., Duboue,P.A. and Rzhetsky,A. (2001) Disambiguating proteins, genes, and RNA in text: a machine learning approach. Bioinformatics, 17 (Suppl. 1), S97–S106.

Hirschman,L., Park,J.C., Tsujii,J., Wong,L. and Wu,C.H. (2002) Accomplishments and challenges in literature data mining for biology. Bioinformatics, 18(12), 1553–1561.

Huang,M., Zhu,X., Hao,Y., Payan,D.G., Qu,K. and Li,M. (2004) Discovering Patterns to Extract Protein-Protein Interactions from Full Texts. Bioinformatics, 20(18), 3604–3612.

Humphreys,K., Demetriou,G. and Gaizaukas,R. (2000) Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. Proceedings of Pacific Symposium on Biocomputing, 502–513.

Jenssen,T.K., Laegreid,A., Komorowski,J. and Hovig,E. (2001) A literature network of human genes for high-throughput analysis of gene expression. Nat Genet, 28, 21–28.

Kanehisa,M., Goto,S., Kawashima,S. and Nakaya,A. (2002) The KEGG databases at GenomeNet. Nucleic Acids Res., 30, 42–46.

Kim,J., Ohta,T., Tateisi,Y. and Tsujii,J. (2003) GENIA corpus—a semantically annotated corpus for bio-textmining. Bioinformatics, 19 (suppl. 1), i180–i182.

Krauthammer,M., Rzhetsky,A., Morozov,P. and Friedman,C. (2000) Using BLAST for identifying gene and protein names in journal articles. Gene, 259, 245–252.

Marcus,M.P., Santorini,B. and Marcinkiewicz,M.A. (1994) Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics, 19, 313–330.

Ng,S. and Wong,M. (1999) Toward Routine Automatic Pathway Discovery from On-line Scientific Text Abstracts. Genome Informatics Workshop 1999, 104–112.

Ono,T., Hishigaki,H., Tanigami,A. and Takagi,T. (2001) Automated extraction of information on protein–protein interactions from the biological literature. Bioinformatics, 17(2), 155–161.

Proux,D., Rechenmann,F., Julliard,L., Pillet,V.V. and Jacq,B. (1998) Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. Genome Inform. Ser. Workshop Genome Inform, 9, 72–80.

Pustejovsky,J., Castano,J., Zhang,J., Kotecki,M. and Cochran,B. (2002) Robust relational parsing over biomedical literature: extracting inhibit relations. Proceedings of Pacific Symposium on Biocomputing, 362–373.

Ramani,A.K., Bunescu,R.C., Mooney,R.J. and Marcotte,E.M. (2005) Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. Genome Biology, 6(5), R40.1–11.

Rindflesch,T.C., Hunter,L. and Aronson,A.R. (1999) Mining molecular binding terminology from biomedical text. Proc. AMIA. Symp., 127–131.

Rindflesch,T.C., Tanabe,L., Weinstein,J.N. and Hunter,L. (2000) EDGAR: extraction of drugs, genes and relations from the biomedical literature. Proc. Pac. Symp. Biocomp, 517–528.

Rzhetsky,A. *et al*. (2004) GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. Journal of Biomedical Informatics, 37, 43–53.

Sekimizu,T., Park,H.S. and Tsujii,J. (1998) Identifying the Interaction between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts. Genome Informatics Workshop, 62–71.

Tanabe,L. and Wilbur,W.J. (2002) Tagging gene and protein names in biomedical text. Bioinformatics, 18, 1124–1132.

Temkin,J.M. and Gilder,M.R. (2003) Extraction of protein interaction information from unstructured text using a context-free grammar. Bioinformatics, 19(16), 2046–2053.

Thomas,J., Milward,D., Ouzounis,C., Pulman,S. and Carroll,M. (2000) Automatic Extraction of Protein Interactions from Scientific Abstracts. Proceedings of the 5th Pacific Symposium on Biocomputing, 541–552.

Wong,L. (2001) PIES, a protein interaction extraction system. Pacific Symposium on Biocomputing, 520–531.

Yu,H., Hatzivassiloglou,V., Friedman,C., Rzhetsky,A. and Wilbur,W.J. (2002) Automatic extraction of gene and protein synonyms from MEDLINE and journal articles. Proc. AMIA. Symp., 919–923.

Xenarios,I., Rice,D.W., Salwinski,L., Baron,M.K., Marcotte,E.M. and Eisenberg,D. (2000) DIP: The database of interacting proteins. Nucleic Acids Res., 28, 289–291.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e227–e235
doi:10.1093/bioinformatics/btl255

Learning MHC I—peptide binding

Nebojsa Jojic[1,†,*], Manuel Reyes-Gomez[1,†], David Heckerman[1], Carl Kadie[1] and Ora Schueler-Furman[2]

[1]Microsoft Research, Redmond WA 98052, USA and [2]Dept. of Molecular Genetics and Biotechnology Hadassah Medical School, The Hebrew University of Jerusalem, Israel

ABSTRACT

Motivation and results: Motivated by the ability of a simple threading approach to predict MHC I—peptide binding, we developed a new and improved structure-based model for which parameters can be estimated from additional sources of data about MHC-peptide binding. In addition to the known 3D structures of a small number of MHC-peptide complexes that were used in the original threading approach, we included three other sources of information on peptide-MHC binding: (1) MHC class I sequences; (2) known binding energies for a large number of MHC-peptide complexes; and (3) an even larger binary dataset that contains information about strong binders (epitopes) and non-binders (peptides that have a low affinity for a particular MHC molecule). Our model significantly outperforms the standard threading approach in binding energy prediction. In our approach, which we call adaptive double threading, the parameters of the threading model are learnable, and both MHC and peptide sequences can be threaded onto structures of other alleles. These two properties make our model appropriate for predicting binding for alleles for which very little data (if any) is available beyond just their sequence, including prediction for alleles for which 3D structures are not available. The ability of our model to generalize beyond the MHC types for which training data is available also separates our approach from epitope prediction methods which treat MHC alleles as symbolic types, rather than biological sequences. We used the trained binding energy predictor to study viral infections in 246 HIV patients from the West Australian cohort, and over 1000 sequences in HIV clade B from Los Alamos National Laboratory database, capturing the course of HIV evolution over the last 20 years. Finally, we illustrate short-, medium-, and long-term adaptation of HIV to the human immune system.

Availability: http://www.research.microsoft.com/~jojic/hlaBinding.html

Contact: jojic@microsoft.com

1 BACKGROUND AND DATASETS

The development of computational methods that predict protein folding and binding is of considerable interest to the scientific community. In addition to furthering our understanding of basic chemical-physical principles that govern the complexity of protein structure, results in this area may also lead to important medical applications. Current research in this area focuses on complex physics-based models using a large number of particles to describe not only the proteins, but also the solvent molecules that surround them.

This paper is motivated by the following observation: Protein binding is at heart of many biological processes which have been heavily studied at a higher level, and so a number of studies have provided indirect sources of information that could be mined to infer unknown parameters of a physics-based binding model. For example, many of the binding configurations bear significant similarities, and therefore the known structures of representative protein complexes can be useful in inferring geometry of binding or binding strength for new proteins. In addition, in some cases, there are publicly available datasets of experimental binding energies (or affinities) for mutated proteins and certain molecules. Other biological experiments are concerned only with the result of a binding process within a more complex system, and so their results may provide only binary information (whether or not the proteins of interest bound or not in a specific context). If interpreted jointly, these diverse sources of data could significantly contribute to our understanding of a system, improve our ability to predict binding partners, and may eventually allow us to manipulate interactions of interest.

Here we focus on one example of such joint data interpretation grounded in a simple physics-based binding model whose purpose is the prediction of the binding energy of peptides to Major Histocompatibility Complex (MHC) class I molecules. MHC class I molecules participate in the detection of foreign proteins expressed within cells. Proteins in the cell are processed to peptides of 8-11 residues length, and some of them are loaded onto MHC molecules which travel to the cell surface and present them to other components of the immune system. In particular, presented peptides may be recognized by cytotoxic T cells, which can destroy the cells deemed to be operating improperly because they present unexpected MHC-peptide complexes. The ''unusual'' complex can be formed as a consequence of a variety of events, such as cell damage, mutation (e.g., cancer), or viral infection, and more recently, organ transplantation.

Due to the importance of this process, it has been experimentally studied in a variety of ways. We describe how we used these studies to train a novel adaptive double threading model of MHC-peptide binding which does not only point out peptides with very low binding energies (good binders, or potential epitopes), but also ranks the peptides with intermediate levels of binding. Adaptivity and double threading make our model appropriate for predicting binding for alleles for which very little data (if any) is available beyond just their sequence, including prediction for alleles for which 3D structures are not available. Armed with this tool, we

*To whom correspondence should be addressed.
†The first two authors contributed equally.

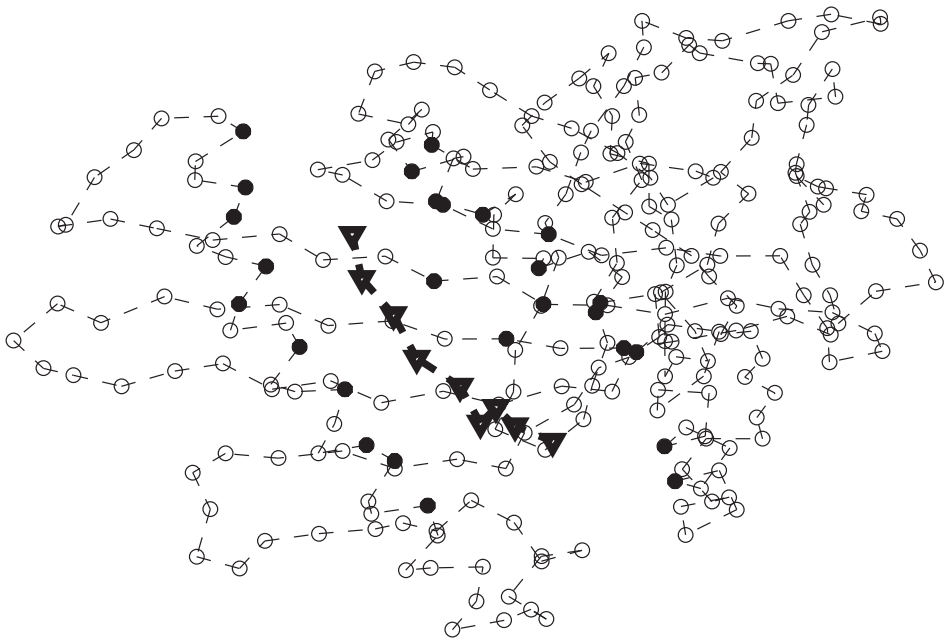


Fig. 1. 3D structure ofMHCA0201 bound to peptide GILGFVFTL (PDB code 1hhi; Madden *et al*., 1993). The centers of the peptide residues are marked in 3D space by triangles and the centers of MHC's residues are marked by circles. Residues in the peptide binding groove of the MHC (i.e. within 4A of the peptide) are marked by filled circles. The lower panel shows the MHC A0201 sequence, with groove sites indicated by increased font size.

are able to demonstrate the effects of immune pressure on HIV sequence evolution within a host and on a population level.

To train the model we can use the following types of data:

MHC I sequence data. We focus here on human MHC class I molecules: they are encoded in three regions of the human genome, labeled A, B, and C. These regions are among the most variable in the human genome, with dozens to hundreds of different MHC variants in each region. Since each individual inherits genes from two parents, each of us has at least three and up to six different MHC molecules operating in our cells. As different MHC molecules typically bind to different peptides, it has been very important to immunologists to classify MHC types. For example, organ transplant recipients may reject organs of donors with non-matching MHC types, as the cells in these organs will present MHC-peptide complexes that are new to the immune system of the recipient. Modern MHC typing is performed by sequencing, and the sequence data for all known MHC variants is available.

MHC-peptide complex structure data. The importance of peptide-MHC interactions to the immune response has motivated crystallographers to solve the structures of a range of different MHC-peptide complexes. In several cases, the structural variability of a specific MHC allele could be assessed by solving the structure of this allele when bound to a range of different peptides. An example of such a MHC-peptide complex structure and sequence is given in Fig. 1.

The present study is based on a dataset of 37 different MHC-peptide complex structures that was also used by (Furman *et al*., 2000). The structures were downloaded from the RCSB protein data bank (Berman *et al*., 2000: http://www.rcsb.org/pdb/).

MHC-peptide binding affinities. The relative binding ability of different peptides to a specific MHC molecule can be directly assessed by competition experiments. The peptide concentration that leads to 50% inhibition of a standard peptide, IC50, is measured, and the relative binding energy can be described as the ratio between the IC50 of the standard peptide and that of a test peptide (Sette *et al*., 1994). The result of such experiments is a set of *relative* binding energies (negative logarithms of the relative concentrations), for different MHC-peptide combinations. This study used a dataset of 870 different combinations from Furman *et al*., 2000, which capture a large range of different binding energies, as discussed in Section 3.

Known good binders (epitopes) and nonbinders. Viral or cancer epitopes, and other excellent binders are often discovered by EliSPOT essays that capture the reaction between T-cells of exposed patients with peptides containing suspected epitopes. Other peptides are known to evoke only very low reactivity in binding essays. Large databases of known epitopes, as well as nonbinders, for various MHC molecules are publicly available. We have used the SYFPEITHI database (Rammensee *et al*., 1999: http://www.syfpeithi.de/), Los Alamos National Laboratory

HIV Database (http://www.hiv.lanl.gov/) and the MHCBN Database (Bhasin *et al*., 2003: www.imtech.res.in/raghava/mhcbn/third.html). These databases provided us with ''binary'' energy data for many peptides (by simply indicating if a peptide is a strong binder, or a non-binder with respect to a particular MHC type).

2 THE BINDING ENERGY MODEL

Our binding energy model is based on the geometry of MHC-peptide complexes, and is motivated by the *threading* approach (Jones *et al*., 1992). Its implementation in (Furman *et al*., 2000) is here augmented by including learnable parameters. We demonstrate that these parameters can be estimated by using all of the described data jointly.

In general, threading aims at evaluating the compatibility of a certain protein sequence with a certain protein structure: The sequence is threaded onto the structure, and a list of contacting amino acid pairs is extracted, based on contacting residue positions (defined as residues in close proximity, e.g. that have at least one pair of atoms less than 4.5A apart). In order to allow estimation of the binding energy of any peptide with an MHC molecule whose structure in complex with some other peptide is known, we assume that the proximity pattern to the peptide in the groove does not change dramatically with the peptide's sequence.

Assuming that energy is additive, and that the pairwise potentials depend only on the amino acids themselves—and not on their context in the molecule—the energy becomes a sum of pairwise potentials taken from a symmetric 20×20 matrix of pairwise potentials between amino acids. These parameters are computed based on the aminoacid binding physics, or from statistical analyses of amino acid pair contact preferences in large sets of available protein structures. Several sets of pairwise potentials have been described in the literature, each derived in a different way (for review see Melo *et al*., 2002). Obviously, the choice of pairwise potential matrix can dramatically alter performance of the energy predictor (Furman *et al*., 2000).

The advantage of the original threading-based approach lies in its independence on binding data. In this approach, as long as a structure of the MHC-peptide complex is available, an allele can in principle be characterized without the need of multiple tedious binding experiments. However, the very same data used for verification of the original threading approach could be used to refine it in a data-driven way. Furthermore, over the last few years a large amount of additional data about binding peptides has been produced for a range of different alleles. Combining the threading approach with a machine learning philosophy of fitting to data, we show that it is possible to estimate a pairwise potential matrix and also learn additional parameters that make the results less sensitive to approximations made in the original threading model.

In order to motivate the parameterization of our model, we start with a slightly more general mathematical definition of the basic threading model, which predicts the binding energy E as a function of the structural template m, the MHC sequence $\mathbf{s}$, and the peptide sequence $\mathbf{e}$, as

$$E(m,\mathbf{s},\mathbf{e}) \approx \sum_i \sum_j \phi_{\mathbf{s}_i,\mathbf{e}_j} h(d^m_{i,j}), \tag{1}$$

where i and j are sequence positions in the MHC molecule and the peptide respectively, ϕ are the pairwise potentials discussed[1], and $d^m_{i,j}$ is the distance between the i-th MHC residue and the j-the peptide residue in the m-th 3D structure (as we have different structures for different molecules)[2]. Finally, in the threading approach, function h is simply the step function

$$h(d) = \begin{cases} 1, & d \le d_{thr} \\ 0, & d > d_{thr} \end{cases}. \tag{2}$$

The threading model is based on the rational approach, which uses physical models to predict the binding energy for a new MHC-peptide complex when a crystal structure (indexed by m) and the sequence of both the MHC molecule and the peptide ($\mathbf{s}$ and $\mathbf{e}$, respectively) are given.

In order to use the abundant direct or indirect information about binding to improve the threading model, and to allow reliable predictions even in the absence of the known structural templates, we make a few adjustments to this model. First, we consider parameters ϕ as hidden variables, with the previously published pairwise potential matrix serving as a basis of the prior on ϕ to avoid over training. Second, instead of the step function, we use a soft step (sigmoid),

$$h(d) = \frac{1}{1+e^{-a(d-d_{thr})}} \tag{3}$$

increasing the robustness of the predictor to slight variations in the geometry of the structural model (residue pairs with a close-to-threshold distance might suddenly be turned off if the distance is only slightly above the threshold). The parameters of h can be learned, setting the threshold (d_{thr}) and the softness a of the step. Finally, we add weights w^m_j to allow our model to adapt to the errors introduced by the strong assumption that all close residue pairs (as defined by h) will contribute to the energy independently. A stringent threshold parameter will produce a very sparse set of pairs i, j that contribute to the energy, and in this case each pair can be assumed to contribute independently. However, many important interactions might be missed by applying a stringent threshold. A loose threshold on the other hand will result in the inclusion of non-relevant residue pairs (amino acid pairs that in fact do not significantly interact in the structure). Including these additional contributions into the energy function might blur the signal. In addition, residues will likely interact simultaneously with several neighbors, which could question the additive model.

In order to address this problem, we add MHC-specific weights $w^m_{i,j}$ to the threading equation with altered function h:

$$E(m,\mathbf{s},\mathbf{e}) \approx \sum_i \sum_j w^m_{i,j} \phi_{\mathbf{s}_i,\mathbf{e}_j} h(d^m_{i,j}), \tag{4}$$

For these weights we use a Gaussian prior favoring $w^m_{i,j} = 1$. The model is designed so that it reduces to standard threading when priors are strong enough to ignore the dataset of energies E for various peptide and MHC combinations. However, in our experiments the priors are left weak enough so that the data can dominate the learning process, and the priors simply serve as a measure against over-fitting.

Note that several variants of the model can be derived from this basic form, depending on how many parameters we want to

[1] ϕ is a 20×20 matrix of potentials for different pairs of amino acids.

[2] In fact, the 3D structure of a MHC-peptide complex may vary slightly for different peptides, in which case a consensus distance is used. See Furman *et al*., 2000 for details

train and how much data we have. For instance, the weights w can be shared across all MHC types, leaving only the sequence s of a molecule to define its behaviour in the model. Furthermore, a single consensus geometry for all types can be used, removing index m from the model completely. Another way of reducing $E(m, \mathbf{s}, \mathbf{e})$ to simply $E(\mathbf{s}, \mathbf{e})$, is to treat structure index m as a hidden variable and infer it with help of proper priors, sequence similarity, or in cross-validation during training. The simpler variants, more capable of generalization, are especially interesting when the goal is prediction of binding energies for new alleles for which no binding data is available. In fact, all forms of this model are based on a physics-based approach which primarily uses the protein sequences into account when evaluating the binding affinity. The MHC type is not primarily captured by its symbolic name (e.g., A0201), but by its sequence as shown in Fig. 1. Thus, applications beyond epitope or energy prediction for each molecule in isolation are possible, e.g., studying the effect of MHC mutations on the efficacy of the immune system in different infections.

We assume Gaussian noise in the energy data (perhaps there are better models motivated by the physics of the process), and we fit the model by standard variational learning, which is needed because of the bilinear dependence of E on ϕ and w. As the optimization criterion becomes quadratic (ignoring parameters of h for a moment), the variational inference essentially iterates between a linear regression to find ϕ variables (penalized appropriately by the prior) and a regression that estimates weights w, again taking into account the Gaussian prior favoring $w_{i,j} = 1$. Refinement of step function parameters (d_{thr} and a) is interleaved with these two steps. For MHC molecules for which we do not have the 3D structure on which to define $d_{i,j}$, we use the available structure of a related MHC molecule with the highest sequence similarity. This is motivated by the fact that across all MHC molecules, the geometry of the groove (i.e. the residues that are in proximity of the peptide) does not change significantly, even when the amino acid content is significantly different.[3] If we view this model as generative, then m can be considered as a hidden variable influencing the sequence $\mathbf{s}$, thus allowing inference of m from $\mathbf{s}$. In principle, in inference of m, both $\mathbf{s}$ and d should be taken into account, but we avoided that in our initial experiments for simplicity. The prior parameters can be tuned through cross validation on the training set.

The dataset of binding energies can be directly used in training our model, but the dataset of known good binders and non-binders requires a treatment of missing energy values. We simply used the lowest binding energy in the binding energy dataset for good binders (epitopes), and similarly, the highest binding energy for the non-binders. Alternatively, the spread between the binding energies of the binders and non-binders can be maximized, or a cost function different than quadratic can be used which punishes bad but not good binding energies for good binders, and does the opposite for non-binders.

It is important to note that we fit all MHC-peptide complexes together, as ϕ parameters are shared across all data. The w_{ij}^m parameters, on the other hand, are specific to a particular MHC geometry (obtained by crystallography). Joint training helps energy prediction for individual MHC types (training only on a limited number of MHC molecules degrades the performance of the predictor on the test data even for the MHC molecules *included* in training). Also note that the model is set up so that it would provide an energy prediction after training even for MHC molecules for which no data other than their sequence is given. The ϕ parameters estimated from the existing data would then be used together with uniform weights $w_{ij}^m = 1$, as dictated by the prior.

[3]In fact, different MHC molecules align well and only 10% of the residues show sequence variability. The ''groove'' residues, however, are the most variable with about 30% of them showing sequence variability, even between two molecules coded in the same region of the genome (A, B or C).

Table 1. Summary of the IC50 dataset used in Sect. 3

	Good binders	Intermediate	Non binders
A0201, peptide length 9	62	254	202
A0201, peptide length 10	27	138	100
A6801, peptide length 9	21	74	35
B2709, peptide length 9	11	11	44

3 MODEL PERFORMANCE ON DIFFERENT TYPES OF DATA

In this section, we empirically illustrate how the model behaves in different situations, such as the usage of binary and/or continuous energy data, with different training set sizes and MHC compositions.

The experimental binding energies (or equivalently IC50 ratios, whose negative log corresponds to energy) for peptides in the set used in this section covered a large range, with only some of the peptides having very low energies (epitopes). To illustrate, we divide peptides into three categories: good binders (IC50 ratio >0.1), non-binders (IC50 ratio <0.0001), and intermediate binders with values in between, as suggested by Furman *et al*., 2000. Table 1 summarizes the data in terms of the MHC molecules, peptide lengths and the binding strength.

In order to compare our method to standard threading, we report the performance of our predictor in terms of peptide ranking measured by Spearman correlation factor, as proposed by Furman *et al*., 2000. This measure varies between -1 and 1, with values close to one indicating that sorting the peptides by their predicted energies produces a similar ranking as sorting by the experimentally measured energies. In a first step, we verified that the numbers obtained by the original threading approach (Furman *et al*., 2000) could be reproduced. In contrast to the threading approach, the method presented here requires training, and for this purpose, the data was divided 100 times into random training/testing partitions (70% for used for training, with the data distribution for both sets kept similar to the above table), and we report the average performance, as well as the variance across the experiments.[4]

Table 2 indicates that our model outperforms the threading model when the direct and indirect information about MHC binding is used to train the model.

Note that for our model the potentially most influential type of data are binding energy measurements (i.e. IC50 values), but this

[4]Threading approach, on the other hand, is rational, not data-driven and so it uses no training data and provides a single number as an output.

Table 2. Comparison of the standard threading and the trained bilinear model

	Threading	Bilinear model	Standard deviation
A0201, 9mers	0.57	0.78	0.03
A0201, 10mers	0.61	0.82	0.03
A6801, 9mers	0.20	0.67	0.13
B2705, 9mers	0.39	0.71	0.09

Table 3. The ability to predict binding for one type by training on other two (transfer)

	Full transfer	Partial transfer
A0201	NA	0.6067 (196 + 869)
A6801	0.23 (584)	0.2974 (584 + 869)
B2705	0.33 (648)	0.5958 (648 + 869)

Full transfer refers to the use all the available training data (continuous and binary) for two MHC types and predicting binding on the third based on its sequence. Partial transfer refers to using all available data for two types as well as the binary energies (but not continuous) of the third type to predict binding energies in the test set for the third type. The results are quantified in terms of Spearman correlation factor between predicted and true binding energies. The numbers in parenthesis are the numbers of training samples (continuous+binary) in different experiments. Full transfer for A0201 could not be performed as removing all A0201 data did not leave enough data for training. See Sect. 5 for results on larger datasets.

kind of data is scarce and is not available for many protein binding problems (but see Sect. 5 about the recent availability of this data for some MHC types). It is therefore of interest to investigate whether the present approach could also be applied to MHC types not experimentally tested in this way, by using information from related, experimentally scrutinized alleles. For this purpose, we evaluate the ability of our model to predict binding energies when some types of training data are not available for MHC types of interest. For each of three MHC types (A0201, A6801 and B2705), two models of nonamer binding were trained: the first using only the experimental binding energies for the remaining two MHC molecules (simulating the situation where the peptide binding to a new MHC allele is modeled), and the second using both the experimental binding energies for the remaining two MHC molecules and 869 binary energies for all three MHC types (simulating the situation where binary data is available for the allele of interest, e.g., through related research, such as epitope discovery, or tracking evolution of a pathogen; but the direct IC50 experiments are not available). In all cases, of course, the test set of known binding energies, was unrelated to the training data. These experiments are summarized in Table 3 and they illustrate how much the peptide preference of a particular allele can be characterized by including binding data for other MHC alleles.

As can be seen, without the information about the specific allele in the training set (column 1), the performance is reduced to values similar to the original threading approach, highlighting the significant contribution of this source of information (compare to Table 2). Note that this experiment could not be performed for A0201 due to insufficient data (around 100 examples, whereas just the number of parameters in the potential matrix is over 200). On the other hand, addition of binary energies from the alleles significantly improved the prediction (see column 2), indicating good generalization capabilities of the model. It is important to note that this experiment was performed on a small dataset in order to study the effects of prior knowledge (3D structure, MHC sequence, and threading model) as well as the value of binary data. In the next section, we revisit the issue of predicting binding for an allele based only on its sequence and the IC50 data for other alleles, but this time using much more data that recently became available.

In order to further evaluate the performance of our method on the data for which only binary energies are known, we used the whole set of binding energies in Table 1, all available 3D structures (for inference of m, when the structure of an MHC molecule is not known), and some of the binary data for training, leaving the rest of the binary data for testing. Again, the training and testing sets are chosen randomly 10 times, and both average performance and the standard deviation are reported. The training set spanned 9 MHC types (A0201, A6801, B2705, A1101, B3501, B5301, A0301, B4402, and B0702), with peptides of lengths 9–10. Since both threading and our method output binding energy, and not a binary decision, we compared the two in terms of ROC curves obtained by varying the good-binder (or epitope) threshold and measuring the number of false positives and false negatives. Our method again significantly outperformed threading (some examples are in Fig. 2), and produced results almost as good as the recently published state of the art in (binary) epitope prediction[5] (Heckerman *et al*., 2006) (more figures available at *www.research.microsoft.com/∼jojic/hlaBinding.html*). Note that for A0301 and B0702 we did not have crystal structures, and yet, our adaptive double threading approach was able to adequately predict peptide binding based on the known sequence of the allele, and a structure of a related allele. Additional examples of predictions based on structures of related alleles, compared to predictions based on the actual crystal structure are available at the above web site.

While the results in this section indicate that the use of binary data is justified, we should point out the important caveat. The epitope data in literature comes form different sources, and some ways of experimentally discovering epitopes do not capture only MHC binding but also other processes that lead to immune reaction (e.g., cleavage and T-cell binding). This means that any tunable model, including ours, when trained on lots of binary data, may capture some of these other effects, becoming better at predicting known epitopes, but worse in predicting strictly MHC binding. At the same time, the constrains in the model structure make our model more suited to modeling IC50-derived energies, then to general purpose classification, and may thus limit its performance in binary epitope classification, when this classification includes factors other than MHC-peptide binding.

For example, when we trained a recently published epitope predictor (Heckerman *et al*., 2006) on binary data only, we find that this method produces good binary classification results, but without significant correlation of the epitope probabilities with true binding energies for intermediate binders in the test set. On the other hand, the model presented here when trained on the same binary data, still recovers peptide ranking for intermediate binders with statistical significance, but with much less accuracy than is the

[5]Epitope prediction algorithms specialize on binary classification and usually do not predict well the quality of binding for intermediate binders.

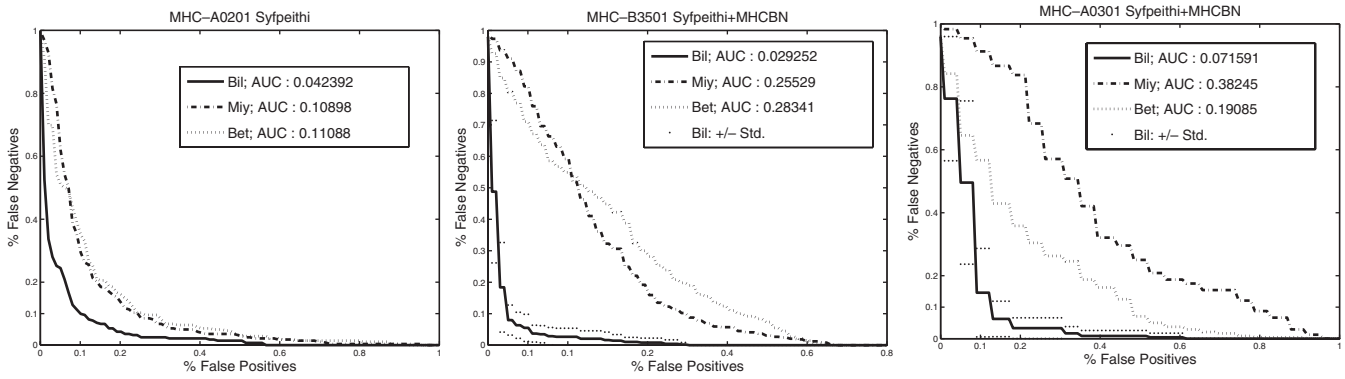


Fig. 2. ROC curves produced by varying the epitope binding energy threshold and computing the number of false positives and false negatives on the SYFPEITHI database. Note that for B3501 only binary energy data was available, while for A0301 no crystal structure was available (The known structure of the MHC molecule with the highest sequence similarity was used, as described in the text.) For the standard threading approach, we used two previously proposed matrices, labeled 'bet' and 'miy' (Furman *et al*., 2000, Betancourt *et al*., 1999, Miyazawa *et al*., 1985), while for our trained bilinear model ('bil'), we also provide standard deviation curves computed over different data splits into training and testing. Over all MHC types, the area under the ROC curve was between 2.5 to 15 times lower for the bilinear model than for either of the threading models.

case when the model is trained on IC50 data. We found that on the binary classification task on epitope data, in comparison to Heckerman *et al*., 2006, the bilinear model suffers a 1% increase in false positive and false positive rates due to its bias towards capturing only the MHC-peptide binding component of being an epitope. This indicates that the tunability of the model makes it possible to tradeoff its energy predictions with its epitope classification capabilities, but it the model may not necessarily extract the single effect (MHC-peptide binding) fully. We are now investigating ways to separate the effects of MHC binding from other effects in binary data and train a combined predictor.

Next, in Sect. 4 we investigate if the predictive power of our model can be used to study the global changes the immune system causes in pathogens, once the model is trained on all available binary and continuous data. Then, in section 5 we evaluate our model on the recently published dataset of IC50 energies.

4 VIRUSES EVOLVE TO MODULATE THEIR BINDING TO MHC MOLECULES

MHC A0201 is one of the most frequent MHC types, especially in the Western world. Using our model, we computed the average binding energy of all HIV 9mers (taking each overlapping peptide from all proteins of the current consensus sequence for clade B) and found it to be equal to 9.74 (the units are of no importance, only the ranking of energies matters). On the other hand, the average binding energy in a randomized HIV is 9.3. The randomized HIV contained the same set of proteins with same lengths but with random aminoacid sequences. The difference in average binding energies has a very strong statistical significance ($p < 10^{-5}$ based on 50 different randomizations), and can be explained by viral evolution—higher average binding energy translates into a smaller total number of presented peptides which trigger immune reaction. Similar patterns should be expected from other viruses, variable enough to use mutation as an escape mechanism. (It is possible that less variable viruses, evolving over a very long time, may still have the same property, and we are planning on investigating this next.)

It has been shown previously that some HIV mutations correlate (weakly) with the MHC types of the host (Moore *et al*., 2002). The binding energy estimators that we developed allow us now to begin to explain these correlations. In Fig. 3, we demonstrate significant correlation ($p < 0.05$) between the average A0201 binding energy and the viral load in the A0201 positive patients from the WA cohort obtained by Moore *et al*., 2002 (as would be expected, in A0201 negative patients we do *not* find any correlation).

For each chronically infected and untreated A0201 positive patient in the cohort, we plot the patient's viral load v.s the sum of 9mer and 10mer average binding energies for A0201 (each patient's HIV was sequenced providing a source of 9mers and 10mers for this computation). The virus whose peptides bind well to a particular MHC molecule is typically under strong immune pressure in patients with this MHC type, and is forced to mutate away from its fittest form towards a form that binds less well to MHC. But, as HIV damages the immune system, the high viral load in the figure indicates a removal of the pressure to escape A0201 binding. Therefore, the negative trend in the figure could be explained by reversion of the viral sequence towards the wild type with higher replicative fitness and lower adaptation to A0201, in patients whose immune system is starting to fail, but other alternative explanations are possible (such as that the intermediate binders in the sequence, become better binders as that serves some purpose to the virus, which after all, infects the immune system). We are investigating these trends further experimentally.

Finally, in Fig. 3, we also track the average binding energy of MHC A0201 to HIV peptides over the last 23 years. The sequences of various proteins from over 1000 patients were obtained from the Los Alamos National Laboratory database. To smooth out the sampling density over time, all sequences were grouped into 3 year time intervals: 1982–1984, 1985–1987, . . . , 2003–2005. The apparent upward trend is statistically weak, but may still indicate that HIV as a population is adapting to the immune systems of the host population. Recently, a trend of HIV fitness attenuation has also been indicated (Arien *et al*., 2005) which would be consistent with this. In order to find out if the trend of modulation

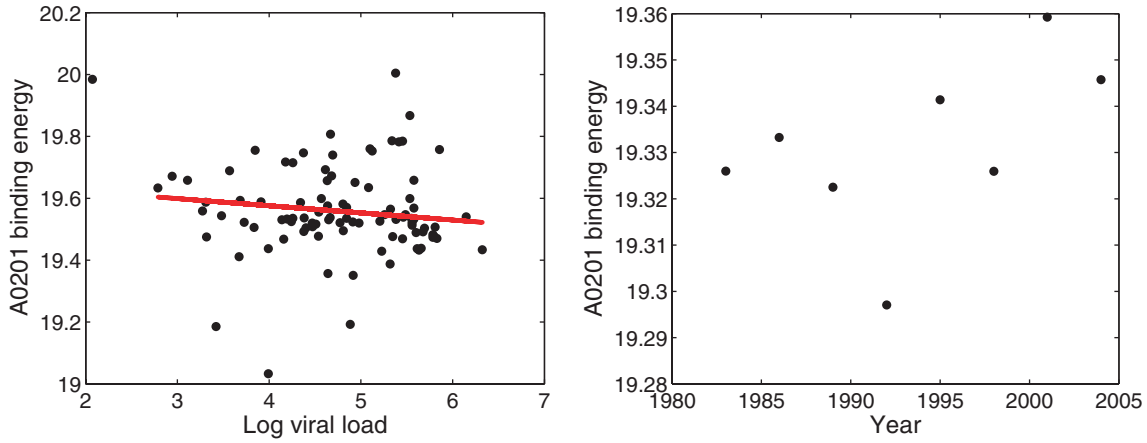


Fig. 3. HIV-MHC A0201 binding energy trends as a function of viral load in individual patients, and the time of sampling.

of MHC binding is significant, we plan to take all MHC alleles into account.

5 PERFORMANCE OF THE MHC-PEPTIDE BINDING MODEL ON A RECENTLY PUBLISHED LARGE DATASET OF IC50 VALUES

Recently, Peters *et al*., 2006 have assembled a large database of experimentally acquired binding energies for a range of MHC molecules. The experiments were performed at the Sette and Buus labs, and Peters *et al*., 2006 go on to also test a variety of published algorithms that can predict MHC-peptide binding. Their published dataset may prove to be the most useful community resource for studying MHC-peptide binding so far. In particular, a very useful feature of their dataset is that the data has been acquired relatively uniformly, with some potential variability due to the fact that the experiments were performed in two different labs.[6] The data consists entirely of IC50 values for 49 different MHC class I alleles, both human and animal. Here, we focus on human alleles from this dataset (the total of 35 A and B alleles), for which the total of 29,371 IC50 values for different MHC-nonamer combinations were tested. Peters *et al*., 2006 show that among the tools available in their labs and on the web, the best performing tool is a neural network proposed by Nielsen *et al*., 2003.

Such a rich dataset and comprehensive comparisons provide several opportunities for additional evaluations of our approach. In particular, as discussed in Sect. 3, the use of heterogeneous data improves overall results, but may skew our model away from solely predicting MHC-binding energy and towards partially capturing additional effects present in the binary experimental data. Nielsen *et al*., 2003 consists solely of the continuous binding energy measurements, and can thus be used to better asses some of the interesting properties of our model. In addition, the amount of data reduces the effects of regularization priors.

[6]In fact, the authors provide a brief analysis of this potential source of error in the paper.

5.1 Predicting binding for new alleles: adaptive double threading

First, we asses the ability of our model to predict binding for a new MHC allele, for which no other data is available but its sequence. This is possible as our trained model performs double threading: not only does it thread a peptide onto the known structure and content of the particular MHC molecule, but it can also use another MHC molecule's structure to thread the new allele's sequence and the peptide on it. When the 3D structure for the allele of interest is available, it is used, but otherwise the best structure from the available database of structures is used (inference of variable m in the model).

To illustrate this empirically, we first focused on the well studied allele A0201, and trained our model on three data subsets and tested the trained models on A0201 test sets in five-fold cross-validation. The first model was trained on IC50 training data for all 35 available molecules in the dataset. The second model was trained on all the data for 34 molecules, but no data whatsoever for A0201 allele, including its 3D structure. The third model is trained on an even more limited dataset which further excluded all A02 types (in this data, A0202, A0203 and A0206), leaving 31 alleles for training. The first model, which was exposed to around 2400 binding energies for A0201 in each fold, achieved the Spearman correlation factor of 0.82, which is comparable to the best result (0.83) reported in Peters *et al*., 2006, and better than all other techniques tested there. However, a more interesting observation is that the second model, which had *no* exposure to A0201 data in training, still predicted A0201 with the Spearman correlation at 0.8, which is only slightly lower than that of the first model. It is important to note that the model did not have the A0201 structure available, and so it could not reduce to standard threading. The model chose to use the structure of the most similar available allele by sequence similarity (A1101). Furthermore, the third model, which had no exposure to any of the A02 types in training, nor the A02 3D structures, still predicted binding with Spearman correlation factor of 0.42. All results are strongly significant with p values virtually zero. Similarly, the Spearman correlation factor for A1101 binding prediction goes from 0.79 to 0.61 when all A1101 data is excluded from training, and only

A1101 sequence is fed to the trained model, but not its structure. This illustrates that our model degrades gracefully as the data related to a particular allele is removed from training, as long as the data for other alleles is available. Therefore, the model is leveraging data for multiple alleles in each of its predictions, and can potentially be used to predict binding for new alleles, given only by their sequences.

5.2 Geometry estimation

The known 3D structure of several alleles is the basis of our bilinear model and it affects the predictions through terms $h(d_{ij}^m)$ in (4). For large pairwise distances d_{ij}, these terms will be virtually zero, thus making the appropriate amino acid pairs irrelevant in prediction. Another way of thinking about the effect of these distances is as a way of regularizing the combined set of parameters $u_{ij} = w_{ij}h(d_{ij}^m)$ in the model $E(m, \mathbf{s}, \mathbf{e}) \approx \sum_i \sum_j u_{ij}^m \phi_{\mathbf{s}_i, \mathbf{e}_j}$, in which distance function is merged with the weights for the pairs. But, if enough data is available, this regularization should not have to be so strongly informed by the structure, and could instead be based on usual norm-regularization. Since our model is grounded in physics, so estimated weights u_{ij} may in fact capture the relevant structure: wherever the distances are large the importance of the pair should be low, and thus the inferred weight should be close to zero.

To test this hypothesis, we selected 8 diverse A alleles and trained the model without the step function h, and with regularization of the norm of u. We limited the pairs i, j only to those that involved variable sites on the MHC molecule (as the conserved sites will have no discriminating effect in training). Then, we compared the learned pairwise weights u_{ij} with the appropriate Euclidean distances d_{ij} between allele and peptide residues in the consensus A0201 structure. Indeed, the Spearman correlation factor between the absolute value of the estimated weights u_{ij} and distances d_{ij} in the 3D structure was negative (-0.16), as expected, and the result is statistically significant ($p < 0.05$). Therefore, by training our model, it is possible, at least to a certain extent, to recover relevant parts of the 3D structure of the binding configuration.

We also note that we have experimented with a simple linear version of the model for binary prediction, which learns directly the products $v_{i,j,\mathbf{s}_i,\mathbf{e}_j} = u_{ij}^m \phi_{\mathbf{s}_i,\mathbf{e}_j}$, without constraining the weights to satisfy a bilinear form. Such a model is forced to learn a weight for any combination of amino acids at any pair of positions in the MHC molecule and the peptide, and is thus vastly over-parameterized. Therefore most of the weights should be equal to zero to avoid over-training. However, we have found that, when nonzero weights are selected using a wrapper method (Kohavi *et al*., 1997), the linear model makes binary predictions as well as the bilinear model, and it also tends to choose i, j pairs with small distances for its nonzero weights, thus performing some structure estimation, as well. We are extending these experiments to the non-binary case.

5.3 Comparison to other techniques

We have also trained our model have trained our model on the nonamers for 35 human alleles on the same folds as Peters *et al*., 2006 and compared with the techniques they analyzed in five fold cross validation. These techniques treat each different MHC allele in isolation from other, which means that they tend to get punished for not using all available data when the allele is not supported by a large amount of training data. On the other hand, when a lot of data for an allele is available, these techniques may have an advantage as they do not have to sacrifice performance on one allele in order to better capture the others and generalize.

Our model achieved an overall test Spearman correlation factor of 0.75, in line with the best performer of Peters *et al*., 2006, which was a neural network proposed by Nielsen *et al*., 2003, and whose Spearman correlation factor on this data was 0.76. In terms of binary classification, the Nielsen *et al*., 2003 beats our model in 18 out of 35 alleles in this data, with our method typically outperforming when the available training data for an allele is small, as would be expected given the ability of our model to generalize over different alleles. Both our model and Nielsen *et al*., 2003 seem to outperform all other techniques compared in Peters *et al*., 2006 by a significant margin. It should be noted again, however, that this data consists of a consistently measured IC50 values for different peptides, and for binary classification tests, only the test data is binarized by thresholding.

The full set of comparisons is available at:
http://www.research.microsoft.com/∼jojic/hlaBinding.html.

6 CONCLUSIONS

We have introduced a new model of MHC-peptide binding, which rather than focusing on binary classification of epitopes, can be used to estimate a high range of binding energies for high resolution MHC types (four digits, based on MHC sequencing). Both in terms of peptide ranking and binary classification performance, our model significantly outperforms the threading model which was the basis of our bilinear model with hidden variables. In individual allele predictions, our model is comparable to the best among the models in the recent comprehensive study (Peters *et al*., 2006). Furthermore, as the model is physics-based there is a potential for its use in settings where the existing models cannot be used. For example, we demonstrated that we can predict binding for new alleles and infer (to a certain extent) the geometry of the binding configuration from binding energy data. The predictive power of our model enabled us to capture HIV evolution patterns in response to the immune pressure of the human hosts (the threading model alone did not show statistically significant trends). We are now investigating medium- and long-term evolutionary response of other pathogens to the pressure created by the cellular arm of the human immune system. The model can also be used to provide binding energies for epitome learning (Jojic *et al*., 2005).

ACKNOWLEDGEMENTS

We thank Corey Moore, Mina John and Simon Mallal, for providing the data from the WA cohort.

REFERENCES

O. Schueler-Furman, Y. Altuvia, A. Sette and H. Margalit, ''Structure-based prediction of binding peptides to MHC class I molecules:Application to a broad range of MHC alleles,'' Protein Science(2000)9:1838–1846.

A. Sette, J. Sidney, MF. del Guercio, S. Southwood, J. Ruppert, C. Dahlberg, H.M. Grey, R. T. Kubo, ''Peptide binding to the most frequent HLA-A class I alleles measured by quantitative molecular binding assays,'' Mol. Immunol.(1994)31:813–822.

M. Bhasin, H. Singh, and G. Raghava, ''MHCBN: A comprehensive database of MHC binding and non binding peptides,'' Bioinformatics(2003)19:665–666.

H. Rammensee, J. Bachmann, N. Emmerich, O.A. Bachor, and S. Stevanovic ''SYFPEITHI: database for MHC ligands and peptide motifs,'' Immmunogenetics (1999)50:213–219.

C. Moore, M. John, I.R. James, F.T. Christiansen, C.S. Witt, and S.A. Mallal ''Evidence of HIV-1 Adaptation to HLA-Restricted Immune Responses at a Population Level,'' Science(2002)296:1439–1443.

C. Yanover, and T. Hertz ''Predicting protein-peptide binding affinity by learning peptide-peptide distance functions,'' Recomb(2005).

K. Arien, R. M. Troyer, Y. Gali, R.L. Colebunders, E. J. Arts, and G. Vanham ''Replicative fitness of historical and recent HIV-1 isolates suggest HIV-1-attenuation over time,'' AIDS(2005)19:1555–1564.

N. Jojic, V. Jojic, B. Frey, C. Meek, and D. Heckerman, ''Modeling genetic diversity with epitomes: Rational design of HIV vaccine cocktails,'' NIPS 2005.

D. R. Madden, D. N. Garboczi, and D. C. Wiley, '' The antigenic identity of peptide-MHC complexes: a comparison of the conformations of five viral peptides presented by HLA-A2,'' Cell(1993)75:693–708.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, ''The Protein Data Bank,'' Nucleic Acids Research(2000)28:235–242.

D. T. Jones, W. R. Taylor, and J. M. Thornton, ''A new approach to protein fold recognition,'' Nature(1992)358:86–89.

M.R. Betancourt, and D. Thirumalai, ''Pair potentials for protein folding: choice of reference states and sensitivity of predicted native states to variations in the interaction schemes,'' Protein Sci(1999)8:361–369.

S. Miyazawa, and R. L. Jernigan R. L., ''Estimation of effective interresidue contact energies from protein crystal structures: Quasi-chemical approximation,'' Macromolecules(1985)18:534–552.

D. Heckerman, C. Kadie, and J. Listgarten, Leveraging Information Across HLA Alleles/Supertypes Improves Epitope Prediction,'' Recomb 2006.

F. Melo, R. Sanches, and A. Sali, ''Statistical potentials for fold assessment,'' Protein Science(2002)11:430–448.

B. Peters, H. H. Bui, S. Frankild, M. Nielsen, C. Lundegaard, *et al*., ''A Community Resource Benchmarking Predictions of Peptide Binding to MHC-I Molecules,'' PLoS Computational Biology(2006) In press. DOI: 10.1371/journal.pcbi.0020065.eor.

M. Nielsen, C. Lundegaard, P. Worning, S. L, Lauemoller, K. Lamberth, *et al*., ''Reliable prediction of T-cell epitopes using neural networks with novel sequence representations,'' Protein Science(2003)12: 1007–1017.

R. Kohavi, D. Sommerfield, and J. Dougherty, ''Data Mining using MLC++, a Machine Learning Library in C++,'' International Journal of Artificial Intelligence Tools,(1997)6:537–566.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e236–e242
doi:10.1093/bioinformatics/btl265

Comparative genomics reveals unusually long motifs in mammalian genomes

Neil C. Jones and Pavel A. Pevzner*

Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA, USA

Between short regulatory motifs and long 'ultraconserved' regions lies a whole spectrum of functional elements that remains uncharted.

– Manolis Kellis, RECOMB Regulatory Genomics satellite workshop, December 2005

ABSTRACT

Motivation: The recent discovery of the first small modulatory RNA (smRNA) presents the challenge of finding other molecules of similar length and conservation level. Unlike short interfering RNA (siRNA) and micro-RNA (miRNA), effective computational and experimental screening methods are not currently known for this species of RNA molecule, and the discovery of the one known example was partly fortuitous because it happened to be complementary to a well-studied DNA binding motif (the Neuron Restrictive Silencer Element).

Results: The existing comparative genomics approaches (e.g., phylogenetic footprinting) rely on alignments of orthologous regions across multiple genomes. This approach, while extremely valuable, is not suitable for finding motifs with highly diverged "non-alignable" flanking regions. Here we show that several unusually long and well conserved motifs can be discovered *de novo* through a comparative genomics approach that does not require an alignment of orthologous upstream regions. These motifs, including Neuron Restrictive Silencer Element, were missed in recent comparative genomics studies that rely on phylogenetic footprinting. While the functions of these motifs remain unknown, we argue that some may represent biologically important sites.

Availability: Our comparative genomics software, a web-accessible database of our results and a compilation of experimentally validated binding sites for NRSE can be found at http://www.cse.ucsd.edu/groups/bioinformatics.

Contact: ppevzner@cs.ucsd.edu

INTRODUCTION

One of the most important decisions the early embryo must make is how to form a central nervous system. Recent studies of this developmental decision led to the Default Model of neural induction that postulated that all ectodermal cells would adopt a neural fate in the absence of intracellular signalling (Munoz-Sanjuan and Brivanlou, 2002). Shortly after the proposal of the Default Model, Chong *et al*. (1995), and Schoenherr and Anderson (1995) discovered a repressor of neuronal specific genes in non-neural cells and characterized the Neuron Restrictive Silencer Element (NRSE) that is the target DNA binding sequence of this repressor (Schoenherr *et al*., 1996). The NRSE motif is somewhat unique in that it is unusually long and has the highest information content among all known vertebrate motifs in TRANSFAC Wingender *et al*. (2001) (with a sufficient number of experimentally confirmed binding sites). Recently, our group (Lunyak *et al*., 2002) and Bruce *et al*. (2004) independently used bioinformatics approaches to extend the small set of experimentally confirmed NRSE sites to a large set of putative NRSE sites in several vertebrate genomes. But without the foreknowledge of NRSE's consensus sequence, could NRSE have been discovered computationally? More generally, if there are other still unknown NRSE-like motifs with unusually high information content, could they be discovered computationally? The recent discovery of the first small modulatory RNA (Kuwabara *et al*., 2004) and its relationship to NRSE implies that the solution of this problem may be important not only in the context of motif finding, but also in the context of finding other smRNAs.

The NRSE motif is very long (20 bp) and conserved (80% identity), which should make it an ideal target for *de novo* motif finding algorithms (e.g., MEME (Bailey and Elkan, 1994)). However, since one knows nothing about which genes an undiscovered motif may regulate, forming an appropriate input sample *a priori* is impossible. Moreover, an instance of NRSE may be millions of nucleotides from the gene that it regulates (Lunyak *et al*., 2002; Schoenherr *et al*., 1996), rendering standard motif search algorithms useless even when coupled with perfectly accurate gene expression analyses.

Recent studies have demonstrated that comparative genomics can overcome the inherent difficulties in searching for transcription factor binding sites (Xie *et al*., 2005; Kellis *et al*., 2003; Lenhard *et al*., 2003). However, most existing comparative genomics approaches rely on phylogenetic footprinting, in which one first constructs alignments between orthologous regions of different genomes and then identifies motifs in these conserved regions. Thus, if the motif to be discovered does not participate in the alignment of the orthologous regions, it will not be discovered. Moreover, even if all of the NRSE occurrences were captured in the alignments, they would still remain undiscovered since most phylogenetic footprinting techniques assume that many instances of a motif within a genome are identical or nearly so (see, e.g., Xie *et al*., 2005). While this assumption holds true (indeed, this assumption is essential) for 6-10 bp transcription factor binding sites, there are hardly any identical instances of the NRSE motif. In fact, out of 22 putative NRSE sites discovered in promoter regions without requiring

*To whom correspondence should be addressed.

alignments, only 9 are found in alignments. The remaining 13 either were aligned with gaps (6) or occur in regions that could not be aligned (7) according to the MLAGAN (Brudno *et al*., 2003) multiple mammalian alignments (human, mouse, rat, dog and chimp).

We believe that a search for motifs of this longer size is important for two reasons. First, cataloguing long motifs in the promoter regions of mammalian genomes may help in determining if the recently-discovered instance of a non-coding RNA transcriptional regulator (Kuwabara *et al*., 2004) is but one of a much larger class of such molecules. The observed effect of adding NRSE dsRNA to an adult neural stem cell is that the cell begins to take on the neuronal characteristics, in part because the protein complex that normally binds to NRSE and behaves as a transcriptional inhibitor of neuron-specific genes becomes a transcriptional enhancer of those genes. Since this operates at the transcriptional level and can enhance gene expression, the mechanism of smRNA must be distinctly different from that of siRNA or miRNA which are both post-transcriptional. Second, the recently discovered juxtaposition of multiple master regulator binding sites (e.g., Oct4 and Sox2) is known to influence the fate of embryonic stem cells (Remenyi *et al*., 2004; Boyer *et al*., 2005) and the combined unusually long binding sequences may be an important signature of combinatorial gene regulation. Conversely, if we deliberately search for long motifs and find nothing, we will have more confidence in the current selection of parameters for motif-finding algorithms.

Below we present a comparative genomics approach that discovers the NRSE motif—along with others whose functions remain unknown—using neither prior information about which genes might be coregulated nor a detailed alignment of orthologous promoter regions. Our results suggest that NRSE is one of several ''long and conserved'' motifs that have been systematically missed by existing comparative genomics approaches (e.g., Xie *et al*., 2005; Ettwiller *et al*., 2005).[1]

Recently, Bejerano *et al*. (2004) discovered long substrings (>200 bp) from vertebrate genomes that were surprisingly well conserved. In this study we discover $\approx$ 20 bp long strings that are surprisingly well conserved across orthologous regions of various mammalian genomes. Like Bejerano *et al*. (2004), we do not speculate as to the function of the motifs we find, but instead provide evidence that they are not statistical artifacts. However, the fact that the NRSE motif appears at the very top of our list is an indication that other motifs in the list may also be functional. Unfortunately, since NRSE is the only known long mammalian motif with such a high degree of conservation, we cannot expect to find other motifs in our list that have known biological roles. A detailed biological analysis of these motifs and the genes they occur near would be a logical next step.

THE COMPARATIVE MOTIF FINDING PROBLEM

An l-mer is a string of length l in the four letter alphabet $\{A, T, G, C\}$. An (l, d)-motif is an l-mer with an associated distance, d, that specifies a maximum allowable number of mismatches. An (l, d)-motif M occurs in a sequence s if there exists a substring in s that is within d mismatches to M or to the reverse complement of M, denoted $\overline{M}$. We may also represent a motif in the alphabet $\{A, T, G, C, N\}$, where N represents a ''don't care'' position. In this case, an (l, d)-motif with t N's can be thought of as a gapped $(l - t, d - t)$-motif where the locations of the t gaps are known.

Suppose we have a family of sequences, $S = \{S_i^j : 1 \leq i \leq n, 1 \leq j \leq m\}$, such that S_i^j represents the ''i-th sequence in species j''. We assume that sequences $S_i^1, \ldots, S_i^m$ in all m species are somehow related, e.g., represent upstream regions of orthologous genes in m species. For a given (l, d)-motif M, let M_i^j be 1 if M occurs in S_i^j and 0 otherwise. One way of framing the traditional motif finding problem (Bailey and Elkan, 1994; Brazma *et al*., 1998) is to search for all M such that $\sum_i \sum_j M_i^j$ is large (e.g., larger than a predefined threshold), though in practice one also imposes a constraint on the information content of the resulting profile. However, the Motif Finding problem loses sight of the relationships between S_i^*, which contains important comparative genomics information about motifs. Instead of $\sum_i \sum_j M_i^j$, we rely on $Score(M, S) = \sum_i \prod_j M_i^j$, in effect forcing the motif to occur in related sequences across all species. When a motif M has a non-zero score, we call it a Π-*motif* in sample $\mathcal{S}$. The Comparative Motif Finding problem is to find all Π-motifs M whose score exceeds a predefined threshold τ.

No efficient algorithms are yet known for the Comparative Motif Finding problem. The exhaustive search approach (see, e.g., Elemento and Tavazoie, 2005) is likely to be too time-consuming for long motifs. Indeed, solutions to the Comparative Motif Finding problem do not necessarily represent sample strings, i.e. strings that appear in some sets S_i^j from S. Nonetheless, finding all sample strings with $Score(M, S) > \tau$ is a simpler problem, and we use an efficient heuristic to solve it.

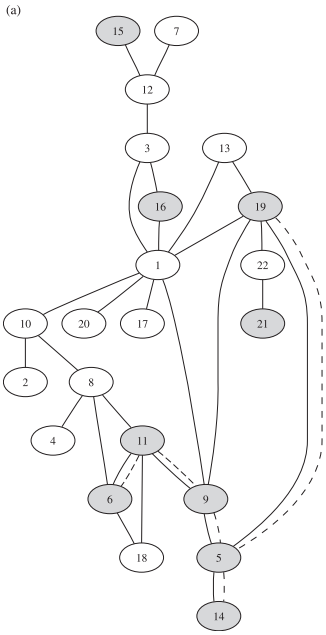
Our approach to solving the Comparative Motif Finding problem is to list all sample strings from one species that represent Π-motifs and cluster the Π-motifs to reveal frequently occuring ones. The algorithm we propose has three basic steps: (i) enumeration, which identifies all Π-motifs corresponding to sample strings; (ii) aggregation, which clusters frequent Π-motifs into a single consensus representation; and (iii) concatenation, which assembles overlapping frequent Π-motifs into a single motif representation. An example of steps (i) and (ii) in the case of the discovered NRSE motif is shown in Fig. 1.

Enumeration proceeds by checking whether each sample string w from S_i^j occurs, with d or fewer mismatches, in each of the strings S_i^* (or $\overline{S_i^*}$).[2] Limiting Π-motifs to sample strings at this stage biases the algorithm towards underreporting motifs; that is, this algorithm will be unable to discover a motif that is overrepresented in the sample but does not explicitly appear in it. However, if this does occur, one would expect some sample string to be an adequate substitute for the ''true'' motif. The algorithm is summarized in Methods and in Fig. 3.

Aggregation takes into account the fact that the enumeration step will rarely discover identical l-mers that represent the same motif due to mutations. Therefore, to discover over-represented motifs we aggregate Π-motifs by performing a clustering procedure on the *similarity graph* whose vertices represent Π-motifs found at the enumeration step. Vertices in this graph are connected by an edge if the Hamming distance between them is no more than

[1]This is not a criticism of existing comparative genomics techniques, but simply a reflection of the fact that they were not designed for the discovery of long motifs.

[2]As one would expect transcription factor binding sites to exhibit few insertions or deletions, the Hamming distance model used here does not account for indels.



(b)

#	Site
1	TTCAGCACCGCGGACAGCGC
2	GTCAGCACCACGGACAGCTT
3	TTCAGCACCGAGGACAGCGC
4	AGCAGCACCGCGGACAGCCC
5	ATCAGCACCACGGACAGCGG
6	GGCAGCACCGTGGACAGCGC
7	TTCAGCACCTGGGACAGTTC
8	GGCAGCACCGCGGACAGCAC
9	CTCAGCACCACGGACAGCGC
10	GTCAGCACCGCGGACAGCTC
11	GGCAGCACCACGGACAGCGC
12	TTCAGCACCGAGGACAGTTC
13	TTCAGCACCGCGAACAGCGG
14	ATGAGCACCGCGGACAGCGG
15	TTCAGTACCGAGGACAGGTC
16	TTCAGCACCGTGGACAGAGC
17	TTCAGCACCGCGGACAGCAT
18	GGCAGCACCAAGGACAGCGC
19	TTCAGCACCACGGACAGCGG
20	TTCAGCACCTCGGAAAGCGC
21	TTCAGCACCACAGACAGTGT
22	TTCAGCACCACGGACAGAGT

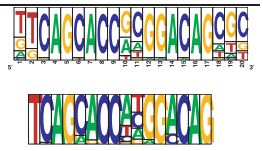


Fig. 1. An example of the motif discovery algorithm as it recapitulates the NRSE motif. Sample strings that are Π-motifs are enumerated from orthologous upstream regions. (a) Similar Π-motifs appear as connected components in the similarity graph. Although the diameter of this connected component is large, the maximum pairwise Hamming distance within the component is small. Consider vertices 6 and 7: the path length between these vertices in the graph is 6, indicating a possible Hamming distance of 12 between the vertices, but the Hamming distance is 6. (b) The consensus sequence of the connected components is shown immediately beneath the table. For purposes of comparison, the motif logo for experimentally determined NRSE sites is shown beneath that. Vertices shown in gray are 9 strings found in MLAGAN mammalian alignments; dashed edges show that the subgraph induced by these vertices comprises four small connected components (the largest one has 6 vertices) instead of one large component on 22 vertices. The remaining strings either had gaps in the MLAGAN alignments or occurred in regions deemed unalignable (i.e., no aligned blocks spanned the region).

$d/2$. Connected components (connected subgraphs) in this graph represent instances of similar Π-motifs. We remark that after aggregations, Π-motifs are no longer constrained to be sample strings.

It turned out that many (l, d) -motifs we discover actually represent parts of slightly longer motifs (this could happen if a binding site is slightly longer than l). In the concatenation step, we connect any two motifs that share significant sequence overlap, thus forming a (possibly) longer motif. Motifs that have a small number (in our application, fewer than 10) of supporting sequences are discarded as not highly overrepresented, and any $5'$ or $3'$ terminal columns in a motif that have fewer than some threshold number of sequences are dropped from that motif, resulting in a motif of some length l' that may be different than l. Afterwards, columns that do not have a clear consensus nucleotide (i.e., at least 50) are labelled as N. Thus, the resulting motif descriptions are not necessarily contiguous (l, d)-motifs in the four letter nucleotide alphabet, but (l', d)-motifs with t gaps, i.e., $(l' - t, d - t)$ gapped motifs.

As an example, consider the *de novo* discovery of a motif with a consensus sequence that is nearly identical to the known NRSE (Fig. 1). The enumeration of (20, 4)-Π-motifs from orthologous upstream promoter regions of genes in human, mouse, and rat results in more than 1 million strings; however, the overwhelming majority of these Π-motifs formed isolated vertices in the similarity graph and were therefore immediately discarded as statistical artifacts. Very few of the remaining connected components had more than 20 vertices. Interestingly, one particular connected component with 22 Π-motifs had a consensus sequence that matched the known NRSE motif. This consensus sequence could then be combined with the consensus sequences from other connected components that are $5'$ and $3'$ shifts of this motif, ultimately leading to a 21 bp motif with 3 ''don't care'' symbols, TNCAGCACCNNGGACAGCGCC. To compare our *de novo* prediction against experimentally validated NRSE sites, we compiled a list of known sites reported in the literature (see Methods); the logo representation of the validated NRSE binding sites is shown in Fig. 1b. Not surprisingly, there was substantial agreement between instances of the predicted motif and experimentally validated NRSE sites. Remarkably, our *de novo* predictions correctly identified two ''wobble positions'' in the middle of the NRSE motif, and also extends the canonical NRSE motif by four somewhat less conserved positions on both the $3'$ and $5'$ ends.

In this study the motif width, l, is set to 20 and the number of allowable mutations, d, to 4. In theory, this algorithm could be used for other values of l and d, though the biologically relevant range of parameters is small. One would expect that the motif width would be less than 30 characters, and d can be chosen accordingly given l so that the expected number of occurrences of an (l, d)-motif would be kept low in the size of the sequence analyzed. Changing the threshold τ represents the trade-off between sensitivity (fewer false negatives) and specificity (fewer false positives).

RESULTS

We applied the above motif discovery algorithm on 5 Kb-long orthologous upstream sequences from human, mouse, and rat. The *de novo* discovery of motifs turned up 606 that were further subjected to statistical tests (see Methods). After filtering, the resulting list contained the 35 motifs shown in Table 1. NRSE appears among the top motifs in this list, thus indicating that our method is indeed capable of finding long motifs in mammalian genomes without prior information about which genes a motif regulates.

Any attempt at *de novo* motif discovery is likely to find some motifs that are functional and many more that are not functional. We approach the problem of distinguishing between these two cases by considering three factors.

First, if the occurrences of a motif are not conserved in the human, mouse, and rat genomes, then that motif is probably not functional. We show that most motifs we find exhibit much higher conservation in all three species than one would expect by random chance, an argument in favor of their functionality.

Second, NRSE is an ''ancient'' motif that is conserved across frog, chicken, and mammals. This implies that the orthologous instances of NRSE motifs in human and rodents (separated by $\approx$ 80 million years of evolution) should be more conserved than the paralogous instances in human that presumably had more time to evolve. Indeed, instances of the NRSE motif exhibit significantly higher conservation between human/mouse/rat genomes (5% divergence on average) than between different instances of the NRSE

Table 1. The significant long motifs found by the algorithm. Motifs that overlap significantly with experimentally confirmed NRSE sites are labelled as such. Columns: Score, the ranking score ($Score(M, \text{G})\text{-np})\sqrt{np}$; # H, the number of hits in human blocks; # HMR, the number of hits in orthologous human, mouse, and rat blocks; inter/intra d_H the inter-species (/intra-species) hamming distance averaged over instances of the motif that occurred in conserved blocks in all three species. The marked instances of the NRSE motif may overlap (e.g., motiffs 17 and 20 overlap by 17 nucleotides)

#	NRSE?	Consensus	Score	# H	# HMR	inter/intra d_H
1	x	GNGNTCAGCACCNCGGACAG	308.2	101	20	1.6/0.7
2		GNGCATNCTGGGANTTGTAG	212.7	154	26	1.6/0.6
3		GCNGCGCGGTCCCTTTAAGA	211.5	92	12	4.7/0.8
4		ANAGGGNTTCTCNCCTGTGTG	211.5	360	97	2.6/1.7
5		GGAGCTGGAGAAGGAGTTNCACTT	201.4	155	23	6.2/1.3
6	x	TNCAGCACCNNGGACAGCGCC	198.6	498	131	2.9/1.2
7		GCNGCCGTTGCCATGGANAC	193.8	157	25	3.1/0.7
8		CCNCGGCGCCGCCATCTTGA	189.2	168	24	4.7/0.9
9		GCGNGGCANTCTGGGANTTGT	182.1	146	20	3.2/1.5
10		CGCCGCCGCCATGTCCGNGG	181.8	229	22	5.0/1.1
11		GCTGGCANCCGCCGCCGCNG	178.2	133	10	3.4/0.7
12		GCNGNGGACTACAACTCCCA	168.0	125	12	3.1/1.3
13		CCNNGGGCGCCGCCATCTTGC	163.5	339	51	4.8/1.0
14		CAGCCAATCAGCGCNCGGCG	162.2	194	20	4.9/1.8
15		CGCGGNGCACGCCGGGAAGC	153.3	208	14	4.7/1.6
16		CTACAANTCCCANAAGGCAC	147.5	222	31	3.4/1.3
17	x	TTCAGCACCANGGACAGCTC	125.4	1078	299	4.7/2.0
18		GCGCTGCAGCCGCTGCNGNG	125.1	203	14	3.4/0.8
19		CCCGCGTCTCCATGGCNACG	123.9	207	17	4.8/1.3
20	x	TNCTTCAGCACCACGGACAG	116.9	688	145	4.5/2.0
21		GCNCAGCCAATCAGCGGGCG	96.6	187	11	4.9/1.8
22		CNTGCTGCNGCGGCCGCCGC	96.3	274	18	2.8/0.8
23		TGCNTTCTGGGAGTTGTAGT	93.4	881	178	4.6/2.1
24		GGCCNCCAGAGGGCGNAGNGG	91.5	214	10	3.4/0.5
25		GACTNCATTTCCCGGCAGGC	91.2	444	44	4.5/1.7
26		GCGCNGCCAATCAGCGCGCGG	88.5	362	28	6.5/1.8
27		CGGCCATGTTTGTNAGGGGC	83.8	183	16	4.6/1.7
28		GNANAAACTACAACTCCCAG	81.7	205	21	3.2/1.5
29		AACTACAATTCCCAGAGNNC	80.9	308	36	2.9/1.0
30		GCCGATTGGCCGCCGCCGCG	80.9	363	18	6.4/1.9
31		CGCGGTGCATNCTGGGACTT	78.9	214	19	4.6/2.1
32		ACANCTCCCGGCAGGCNTCGC	77.9	333	20	5.2/1.6
33		GCCGCCGCCGCNGCNGCTGCNG	77.4	469	31	3.2/2.0
34		ATGTAAATCATATGCAAATG	76.4	3395	991	6.6/5.8
35		GGCCTGGTNGCCATGGCAAC	75.9	624	92	5.1/2.0

motif within the human genome (13% divergence on average). Nearly all of the motifs that we discovered exhibited this property. Such a phenomenon is unlikely for spurious motifs, so this provides another argument in favor of the hypothesis that at least some of the motifs we report are functional.

Third, since the existing repeat masking is imperfect, there is a chance that the motifs we discover are parts of unmasked repeats shared by human, mouse, and rats. While human and rodents share few highly diverged repeats, three of the motifs that we discover represents an *l*-mer from the known repeat families. Thus, one can conclude that the motifs we discover are not parts of unmasked transposable elements.

A common assumption in comparative genomics is that if a motif is functional, then it will be conserved. That is, if our algorithm outputs a sequence motif that does not appear in orthologous sequences more often than can be expected at random (while accounting for the total number of times it occurs in the genome overall) then it can immediately be rejected as noise. However, restricting the definition of conservation to include only bases that are in aligned regions causes unacceptable loss of potentially functional sites for the long motifs that are the focus of this study. Therefore, we define *blocks* (e.g., gene regions) of sequence that are presumably related through evolution without specifying the exact mapping between basepairs. If a motif occurs in the orthologous block in each species, it is considered a conserved instance.[3] Specifically, we extend the region around each gene g (from the list of genes representing orthologous triples) to the interval $[g_{\text{left}}, g_{\text{right}}]$ where g_{left} is the position ''halfway'' between the start of g and the end of previous gene and where g_{right} is the position

[3]This approach only works when the expected number of instances of a motif in a long sequence block is smaller than 1; this holds for (20, 4)-motifs, but it does not hold for shorter motifs, hence the need for alignments in existing studies.

''halfway'' between the end of g and the start of the next gene. For roughly 8% of genes, the resulting intervals $[g_{\text{left}}, g_{\text{right}}]$ turned out to be very long, so we have chosen to trim such intervals to 500 kb from each side, leaving some regions of the genome uncovered by the intervals. The resulting collection of intervals is denoted $\mathcal{G}$ (analagous to $\mathcal{S}$) such that $Score(\mathcal{M}, \mathcal{G})$ is well defined.

We define a score for ranking motifs that is similar to the Motif Conservation Score (MCS) from Xie *et al*. (2005). Assume the motif M appears in b_j blocks in species j and that there are a total of n in each genome. If we randomly mark blocks from each of the m species with probability b_j/n, then the probability of marking any particular block in all m species is $p = (\prod_j b_j)/n^m$. The P-value, or the probability of observing k or more genes that are marked in all m species, is then $1 - \sum_{x=0}^{k-1} F(np, x)$, where $F(a, b)$ is the Poisson distribution with parameter a evaluated at b. However, while a P-value of the ranking score is conceptually more useful than a raw score, it turns out that the P-value usually evaluates to 0 for most of the motifs we report, an indication that the motifs we find are statistically surprising. The expected number of orthologous triples of a motif occurring, according to this naive background model, is np and its standard deviation is approximately $\sqrt{np}$. The ranking score of $(Score(M, \mathcal{G}) - np)/\sqrt{np}$ can be used as a rough estimate of the importance of a motif M.

From the list of 606 motifs we removed motifs that were deemed (a) micro-satellites; (b) occurred more than 10,000 times in the genome; (c) had fewer than 10 conserved hits; or (d) were a variation on A/T-rich patterns like *AAAAAAAAAATTTTTTTTT*. This procedure resulted in 323 motifs that were further investigated to check whether there were motifs in the list that appeared multiple times with minor variations. It turned out that 6 distinct types of motifs appeared multiple times in the list with slightly different or overlapping consensus sequences. These 6 motif families comprised 63 motifs thus reducing our list to 323 − 63 + 6=266 individual motifs. One of the 6 familes corresponded to motifs that were correlated highly with experimentally-determined NRSE binding sites (Sun *et al*., 2005). These motifs originated from six components in the similarity graph whose consensus sequences were sufficiently different to elude the aggregation and concatenation steps of our algorithm. The remaining motifs did not correspond to known transcription factor binding site matrices listed in TRANSFAC (Wingender *et al*., 2001), to miRNA target sequences listed in miRBase (Griffiths-Jones, 2004), to known transposable elements, or to homing endonuclease restriction sites (Roberts *et al*., 2005).

To validate the test for statistical significance of our findings, random substrings of length 20 were selected from the same orthologous set of upstream regions given as input to the motif discovery algorithm. From the set of sampled substrings, some set of columns (between 0 and 4 in total) is selected at random and converted into N characters to account for degeneracy in the motif set. Thus, the randomized ''noise'' motifs consist of strings from the input data set that contain approximately the same pattern of degeneracy as the discovered ''signal'' motifs. The ranking score of the ''noise'' motifs was calculated for motifs that met properties (a)-(c) above. As an aggregate, the scores for the random motifs are statistically different from the scores of motifs output from the motif discovery algorithm (Mann-Whitney rank sum test P-value less than 1×10^{-7}). However, a visual inspection of the box-and-whisker plot of the scores of the two samples (Fig. 2) reveals that while the difference between the sample means may be small, the set of discovered motifs include a large number of outliers (some, but not all, of which correspond to the NRSE motif) that may represent novel biologically functional motifs. Those discovered motifs with ranking score larger than 75 are listed in Table 1. The cutoff score of 75 is conservative because most of the randomly sampled noise motifs with high score were suspiciously similar to poly-A signals, which are systematically conserved and thus not informative.

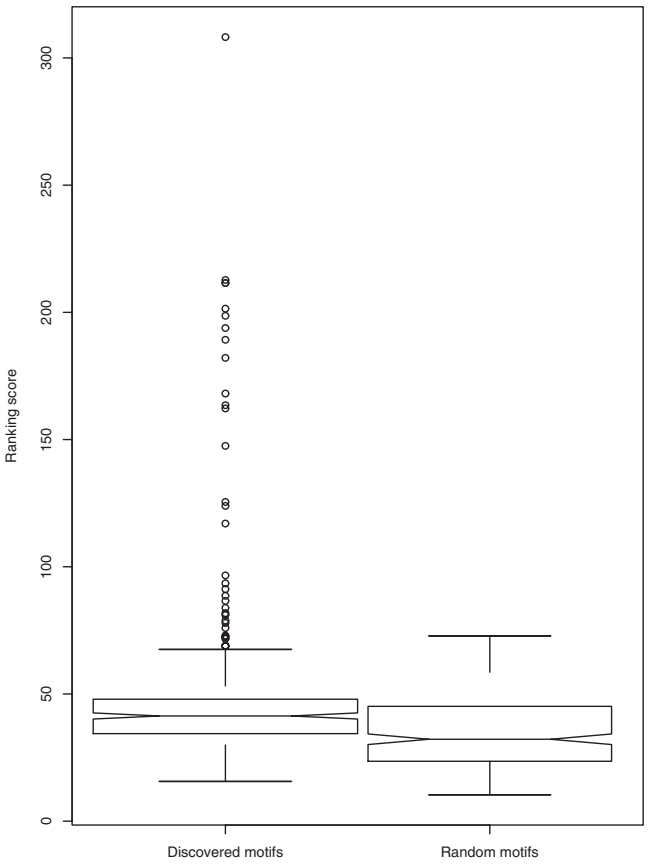


Fig. 2. The distribution of ranking scores for the motifs shows that, while the median score of noise motifs and discovered motifs are different, the overall distributional properties of the two groups are not that different. However, the presence of a number of outliers among the discovered motifs is important: these motifs could be biologically important.

CONCLUSIONS

In one of the first comparative genomics studies, Gelfand *et al*. (1999) discovered a number of conserved strings in bacterial genomes that only later were determined to be riboswitches. Similarly, we have no experimental proof that the strings in Table 1 represent new regulatory elements. However, we have demonstrated that these strings are not statistical artifacts and warrant future experimental analysis. While these computational experiments cannot yet prove whether regulation through smRNAs is a common mechanism in mammalian genomes, they imply that the smRNAs are probably not as ubiquitous as other ncRNAs.

A recent study (Prakash and Tompa, 2005) makes the important point that the assignment of orthology is crucial for comparative genomics approaches. In this study we rely on the publicly available

Input:
Number of species m,
Number of sequences for each species n,
Sets of sequences $\mathcal{S} = \{\{S_1^1, \ldots, S_1^m\}, \{S_2^1, \ldots, S_2^m\}, \ldots, \{S_n^1, \ldots, S_n^m\}\}$,
Distance d,
Motif width l, and
Threshold τ.
Output: (l, d)-motifs M in $\mathcal{S}$ with Score$(M, \mathcal{S}) \geq \tau$.
Enumeration
for each sequence S_i^1:
  for each l-mer s in S_i^1:
    if s occurs in $S_i^2 \ldots S_i^m$ with d or fewer mutations:
      append s to a list V

Aggregation
Create graph G whose vertices are the strings in V
for each pair s and t of vertices in G:
  if $d_H(s,t) \leq d/2$ or $d_H(s,\bar{t}) \leq d/2$:
    add edge (s,t) to G
for each connected component C in G:
  append consensus sequence of C to list V'

Concatenation
Create graph G' whose vertices are the strings in V'
for each pair s' and t' of vertices in G':
  if s' and t', or s' and $\bar{t'}$ *overlap*: # (See methods)
    add edge (s', t') to G'
for each connected component C' in G':
  form *PWM* p from C' (see Methods)
  discard p if its maximal *support* is less than τ (see Methods)
  discard terminal columns of p that have fewer than 40% of p's maximal support
  compute consensus string of p
    columns with less than 55% majority nucleotide become N
    terminal N columns are discarded
  output consensus

Fig. 3. A pseudocode description of the algorithm. See Methods for a clarification of terms.

mapping of orthologous genes, but acknowledge that we would likely find improved motif predictions if better methods for determining orthology are developed. Our work also extends the recent FastCompare (Elemento and Tavazoie, 2005) algorithm by considering motifs in multiple (rather than pairwise) species and by not limiting the analysis to short motifs as in that study.

The algorithm as described in this study is most suitable for sets of species that can be considered evolutionarily equidistant. We are currently working on extending this algorithm to accomodate more varied phylogenetic relationships (Blanchette and Tompa, 2002).

METHODS

All sequences were repeat masked using the RepeatMasker annotations in the Ensembl sequence database; all annotations and orthology relationships derive from the Ensembl Core and Compara databases, release 32 on the assemblies of *Homo sapiens*, *Mus musculus*, *and Rattus norvegicus* (35e, 34, and 34f respectively). Upstream (5000 bp 5′ of transcription start) genomic sequences from all orthologous gene triplets in the human, mouse, and rat genomes resulting in 14,355 usable sequence regions.

The pairwise Hamming distance among found motifs was computed across (inter) species, and within a (intra) species. Assuming an approximately normal distribution of Hamming distance, the two lists were compared using Student's T-test to determine if the inter species distance was larger than the intra species distance at the 99.9% confidence level. All motifs listed in Table 1 have a significant difference between inter- and intra-species Hamming distance. The higher conservation of the motif within putatively orthologous promoter regions compared to the conservation within nonorthologous positions within a single species may indicate that purifying selection is operating on a portion of that motif's instances.

In the enumeration phase of the algorithm, our method takes a shortcut and arbitrarily chooses one member in each set as a reference sequence (human) and enumerates all l-mers in that sequence such that each of the remaining $m - 1$ sequences in the set contains an l-mer with no more than d mismatches to w or $\bar{w}$. Choosing a reference sequenceintroduces a small bias into the algorithm.

As mentioned above, the length of strings recorded in the Enumeration step is $l = 20$, with a distance of $d = 4$. For efficiency, connected components in the similarity graph with fewer than three l-mers were discarded prior to the construction of the overlap graph used in the Concatenation step. Two l-mers $v_1 v_2 \cdots v_l$ and $w_1 w_2 \cdots w_l$ overlap if there exists an i-suffix of v and an i-prefix of w such that $d_H(v_{l-i} \cdots v_l, w_1 \cdots w_i) \leq d$ where $i \geq 0.8l$. In the application considered here, at least 12 nucleotides were required to match over 16 consecutive positions. Each vertex in the overlap graph corresponds to a connected component in the similarity graph, and therefore represents a potentially large number of enumerated l-mers. The Position Weight Matrix representation was constructed from each connected component in the overlap graph by positioning all related enumerated l-mers in the appropriate columns. This leads to the case where different columns in the PWM have different numbers of contributing sequences, and we refer to that number of l-mers as the support of that column. Any column that has less than 40 of the maximum support within the motif is discarded; as expected, this does not discard any internal columns (which would lead to a motif becoming fragmented). Motifs that had maximum support of less than $\tau = 10$ were discarded as unimportant. Columns that did not have a 51% majority consensus nucleotide were listed as *N*.

The enumeration phase requires negligible memory and time $O(nmL^2)$, where m is the number of species, L is each sequence's length, and n is the total number of sequence regions scanned. The aggregation phase requires, in worst case, time and memory proportional to the square of the number of enumerated strings (which will be much less than nL), and the concatenation phase requires time and memory proportional to the square of the number of connected components from the aggregation phase. In practice, the enumeration phase is run in parallel on a grid and the bottleneck is the aggregation phase which is done on a single computer.

We compare our predicted motifs against experimentally validated NRSE sites that have been reported previously (Schoenherr *et al*., 1996; Sun *et al*., 2005). A total of 48 genes are unambiguously identified in the combined studies, but neither study attempts to identify orthologous sites in multiple species. Of the 31 genes from the mouse genome identified in Sun *et al*. (2005), there are 16 orthologous genes in each of human and rat that also have a substring that matches the consensus string used in that study (*TYAGMRCCNNRGMCAG* with no mismatches). Of the 18 genes in the human, mouse and rat genomes reported in Schoenherr *et al*. (1996), there are 14 orthologous genes in each of the other two species that also have a substring that matches the consensus used in that study (*TTCAGCACCNCG-GACAGNGCC* with 4 mismatches). We combine the set of sites that were confirmed in a lab with the set of sites that are orthologous to sites confirmed in a lab into a database of 167 distinct binding sites across the three genomes. While it is not necessarily true that an orthologous instance of a verified binding site is also a binding site, it seems a safe bet that a large portion of them are. We remark that this database necessarily represents a (presumably small) subset of the biologically active NRSE sites in the genome.

ACKNOWLEDGEMENTS

The authors wish to thank V. Lunyak, M. G. Rosenfeld, and S. Wasserman for many helpful discussions. Computations were performed with the UCSD FWGrid Project, NSF Research Infrastructure Grant Number EIA-0303622.

REFERENCES

Bailey, T. and Elkan, C. (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the International Conference on Intelligent Systems in Molecular Biology*, pages 28–36.

Bejerano, G., Pheasant, M., Makunin, I., Stephen, S., Kent, W., Mattick, J., and Haussler D. (2004). Ultraconserved elements in the human genome. *Science*, **304**, 1321–5.

Blanchette, M. and Tompa, M. (2002). Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Research*, **12**, 739–48.

Boyer, L., Lee, T., Cole, M., Johnstone, S., Levine, S., Zucker, J., Guenther, M., Kumar, R., Murray, H., Jenner, R., Gifford, D., Melton, D., Jaenisch, R., and Young, R. (2005). Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell*, **122**, 947–56.

Brazma, A. Jonassen, I., Vilo, J., and Ukkonen, E., (1998). Predicting gene regulatory elements in silico on a genomic scale. *Genome Research*, **11**, 1202–15.

Bruce, A. W., Donaldson, I. J., Wood, I. C., Yerbury, S. A., Sadowski, M. I., Chapman, M., Gottgens, B., and Buckley, N. J. (2004). Genome-wide analysis of repressor element 1 silencing transcription factor/neuron-restrictive silencing factor (rest/nrsf) target genes. *Proceedings of the National Academy of Sciences, USA*, **101**(**28**):10458–63.

Brudno, M., Do, C., Cooper, G., Kim, M., Davydov, E., Green, E., Sidow, A., and Batzoglou, S., (2003). LAGAN and MLAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Research*, **13**, 721–31.

Chong, J., Tapia-Ramirez, J., Kim, S., Toledo-ARal, J., Zheng, Y., Boutros, M., Altshuller, Y., Frohman, M., Kraner, S., and Mandel, G., (1995). REST: a mammalian silencer protein that restricts sodium channel gene expression to neurons. *Cell*, **80**, 949–57.

Elemento, O. and Tavazoie, S. (2005) Fast and systematic genome-wide discovery of conserved regulatory elements using a non-alignment based approach. *Genome Biology*, 6.

Ettwiller, L., Paten, B., Souren, M., Loosli, F., Wittbrodt, J., and Birney, E., (199). The discovery, positioning and verification of a set of transcription-associated motifs in vertebrates. *Genome Biol*, **6**(**12**), R104.

Gelfand, M., Mironov, A., Jomantas, J., Kozlov, Y., and Perumov, D., (1999). A conserved RNA structure element involved in the regulation of bacterial riboflavin synthesis genes. *Trends Genetics*, **11**, 439–42.

Griffiths-Jones, S. (2004). The microRNA registry. *Nucleic Acids Research*, **32**, D109–11.

Kellis, M., Patterson, N., Endrizzi, M., Birren, B., and Lander, E. (2003). Sequencing and comparison of yeast species to identify genes and regulatory motifs. *Nature*, **423**, 241–54.

Kuwabara, T., Hsieh, J., Nakashima, K., Taira, K., and Gage, F. H. (2004). A small modulatory dsRNA specifies the fate of adult neural stem cells. *Cell*, **116**, 779–793.

Lenhard, B., Sandelin, A., Mendoza, L., Engström, P., Jareborg, N., and Wasserman, W. W. (2003). Identification of conserved regulatory elements by comparative genome analysis. *J Biol*, **2**(**2**), 13.

Lunyak, V. V., Burgess, R., Prefontaine, G. G., Nelson, C., Sze, S. H., Chenoweth, J., Schwartz, P., Pevzner, P.A., Glass, C., Mandel, G., and Rosenfeld, M. G.(2002). Corepressor-dependent silencing of chromosomal regions encoding neuronal genes. *Science*, **298**, 1747–52.

Munoz-Sanjuan, I. and Brivanlou, A. (2002). Neural induction, the default model and embryonic stem cells. *Nature Reviews in Neuroscience*, **4**, 271–80.

Prakash, A. and Tompa, M. (2005). Discovery of regulatory elements in vertebrates through comparative genomics. *Nature Biotechnology*, **23**:1249–56.

Remenyi, A., Scholer, H.R., and Wilmanns, M. (2004). Combinatorial control of gene expression. *Nat Struct Mol Biol*, **11**, 812–5.

Roberts, R., Vincze, T., Posfai, J., and Macelis, D. (2005). REBASE—restriction enzymes and DNA methyltransferases. *Nucleic Acids Research*, **33**, D230–2.

Schoenherr, C., and Anderson, D. (1995). The neuron-restrictive silencer factor (nrsf): a coordinate repressor of multiple neuron-specific genes. *Science*, **5202**, 1360–3.

Schoenherr, C., Paquette, A., and Anderson, D. (1996). Identification of potential target genes for the neuron-restrictive silencer factor. *Proceedings of the National Academy ofSciences, USA*, **93**, 9881–6.

Sun, Y., Greenway, D., Johnson, R., Street, M., Belyaev, N., Deuchars, J., Bee, T., Wilde, S., and Buckley, N. (2005). Distinct profiles of REST interactions with its target genes at different stages of neuronal development. *Mol Biol Cell*, **12**, 5630–8.

Wingender, E., Chen, X., fricke, E., Geffers, R., Hehl, R., Liebich, I., Krull, M., Matys, V., Michael, H., Ohnhauser, R., Pruss, M., Schacherer, F., Thiele, S., and Urbach, S. (2001). The transfac system on gene expression regulation. *Nucleic Acids Research*, pages 281–3.

Xie, X., Lu, J., Kulbokas, E., Golub, T., Mootha, V., Lindblad-Toh, K., Lander, E., and Kellis, M. (2005). Systematic discovery of regulatory motifs in human promoters and 3' utrs by comparison of several mammals. *Nature*, **7031**, 338–45.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e243–e251
doi:10.1093/bioinformatics/btl259

Distance based algorithms for small biomolecule classification and structural similarity search

Emre Karakoc[1,*], Artem Cherkasov[2] and S. Cenk Sahinalp[1]

[1]School of Computing Science, Simon Fraser University, Burnaby, BC, Canada and
[2]Division of Infectious Diseases, Faculty of Medicine, University of British Columbia, Vancouver, BC, Canada

ABSTRACT

Motivation: Structural similarity search among small molecules is a standard tool used in molecular classification and in-silico drug discovery. The effectiveness of this general approach depends on how well the following problems are addressed. The notion of similarity should be chosen for providing the highest level of discrimination of compounds wrt the bioactivity of interest. The data structure for performing search should be very efficient as the molecular databases of interest include several millions of compounds.

Results: In this paper we focus on the *k*-nearest-neighbor search method, which, until recently was not considered for small molecule classification. The few recent applications of *k*-nn to compound classification focus on selecting the most relevant set of chemical descriptors which are then compared under standard Minkowski distance L_p. Here we show how to computationally design the optimal *weighted* Minkowski distance wL_p for maximizing the discrimination between active and inactive compounds wrt bioactivities of interest. We then show how to construct pruning based *k*-nn search data structures for any wL_p distance that minimizes similarity search time.
The accuracy achieved by our classifier is better than the alternative LDA and MLR approaches and is comparable to the ANN methods. In terms of running time, our classifier is considerably faster than the ANN approach especially when large data sets are used. Furthermore, our classifier quantifies the level of bioactivity rather than returning a binary decision and thus is more informative than the ANN approach.

Contact: cenk@cs.sfu.ca

1 INTRODUCTION

Small molecules (with molecular weights ≤ 500) are very important to the exploration of molecular and cellular functions. They also play key roles in treating diseases: almost all medicines available today are small molecules. Identification of small molecules that are effective at modulating a given biological process or disease state is a fundamental research challenge we are facing today.

Structural similarity search among small molecules is one of the standard tools used in conventional *in silico* drug discovery. Structural similar chemical compounds are usually similar in their physicochemical properties and/or biological activities (Maggiora and Johnson, 1990). Thus, it is common to query small molecules databases with a *probe* compound possessing desirable biological activity to *discover* chemically similar database entries, which would have a higher probability to have the bioactivity of interest. It is also common to perform classification of a compound with an unknown bioactivity level through a similarity search among compounds whose bioactivity levels are known.

This important *ligand-based* drug discovery methodology and classification approach are associated with two fundamental computational problems that need to be addressed. (1) The notion of similarity used in search determines the molecules that are extracted from the database. A notion of similarity which has the highest level of bioactivity discrimination is very desirable and needs to be determined computationally. (2) It is desirable to have efficient algorithms for structural and/or chemical similarity search as the molecular databases of interest include several millions of compounds and linear/brute force search may take significant amount of time (several days in certain large private databases).

Similarity measures for small molecules

Given a notion of similarity among data elements, it is usually possible to obtain a corresponding *distance* measure; searching for structurally most similar molecules to a query molecule in this context corresponds to searching for molecules with the smallest distance to the query molecule. The key premise of this approach is that the notion of a distance is mathematically well defined and algorithms for handling distance based classification, clustering and search are better understood. For example, the search for the most similar molecule to a query compound becomes the Nearest Neighbor Search (NN) problem in the distance domain. This problem is well studied in computer science and a number of efficient algorithms are available for it. This paper, thus, aims to map the above two problems in structural similarity search, i.e. classification and querying, to corresponding problems in nearest neighbor search.

There are various ways to define the descriptors/parameters for the chemical structures stored in electronic collections conventionally used in the modern computer-aided drug discovery (Brown, 1997; Adamson *et al*., 1973).

Such parameters either (1) merely reflect the structural organization of molecules in qualitative manner, such as those used in the popular *structural fingerprints* (employed in NCBI's PubChem database), e.g. the existence of a doubly bonded Carbon pair, a three membered ring, an aromatic atom etc. (MACCS) or (2) reflect various local and global physical-chemical molecular features (chemical descriptors) which are quantitative, such as atomic

*To whom correspondence should be addressed.

weight, aromaticity, hydrophobicity, the number of specific atoms, charge, density, etc. These descriptors serve as independent variables for modern *QSAR* (Quantitative Structure-Activity Relationship) tools including the structural similarity search engines in chemical compound databases.

Given an adequate set of descriptors, it is desirable to have a measure of similarity or alternatively a distance measure under which chemically equivalent molecules have a high level of similarity or small distance, and non-equivalent compounds have a low level of similarity or large distance. The most common measure of similarity amongst sets of molecular descriptors is the so called *Tanimoto coefficient* (Willett *et al*., 1998). Given two descriptor sets (which can be organized in arrays) X and Y, the Tanimoto coefficient is defined to be the ratio of the number of descriptors that are identical in X and Y and the total number of descriptors available for X and Y. The Tanimoto coefficient is in the range [0,1]; a value close to 1 implies similarity and a value close to 0 implies a dissimilarity among the two descriptor sets compared.

Often a collection of descriptors are represented as a bit-vector (e.g. structural fingerprints) where each one of the n possible descriptors is assigned a *dimension*, i.e. natural number between 1 and n (this is the representation used by PubChem and other databases). Let $B(x)$ represent the bit-vector corresponding to a molecule x and let $B(x)[i]$ represent its i^{th} dimension. Given two compounds x and y, the Tanimoto coefficient $T(x, y)$ is then defined as $T(x,y) = (\sum_{i=1}^{n} (B(x)[i] \wedge B(y)[i]))/(\sum_{i=1}^{n} (B(x)[i] \vee B(y)[i]))$.

Although the Tanimoto coefficient provides a measure of *similarity*, it is possible to define a *Tanimoto distance measure* as $D_T(x, y) = 1 - T(x, y)$. Notice that a Tanimoto distance close to 0 implies a Tanimoto coefficient close to 1, i.e. a high level of similarity and a Tanimoto distance close to 1 implies a Tanimoto distance close to 0, i.e. a low level of similarity between x and y.

The Tanimoto coefficient is very popular mostly due to its simplicity. For real valued descriptor arrays (where each dimension has a real value) it is also quite common to use the Minkowski distance of order p, denoted L_p for measuring their similarity. Given two real valued n dimensional descriptor arrays X and Y, their Minkowski distance of order p, namely L_p, is defined as $L_p(X,Y) = (\sum_{i=1}^{n} |X[i]-Y[i]|^p)^{1/p}$. When comparing two structural fingerprints $B(x)$ and $B(y)$, the Minkowski distance of order 1 is equivalent to the well known Hamming distance (see for example (Chen and Reynolds, 2002)): $H(B(x), B(y)) = \sum_{i=1}^{n} |B(x)[i] - B(y)[i]|$.

In order to capture the similarity between compounds more accurately with respect to a particular bioactivity, more sophisticated distance measures can be used. For example, it is possible to assign a relative importance to each structural descriptor in the form of a weight $w_i \in [0, 1]$. The resulting *weighted* Minkowski distance of order 1 can then be defined for two descriptor arrays X and Y as $wL_1(X,Y) = \sum_{i=1}^{n} w_i \cdot |X[i] - Y[i]|$.[1]

[1]To the best of our knowledge all recent studies in this direction show how to assign *binary values* to weights w_i i.e. how to choose the specific descriptors that are most relevant for the application of interest (e.g. (Zheng and Tropsha., 2000; Itskowitz and Tropsha., 2005)). As will become clear later in the paper, we show how to compute optimal *real valued weights* so as to improve the predictive power of our classifier.

Classification methods for small molecules

The descriptor arrays described above can be used for classification of compounds according to a given bioactivity.

One of the most popular classification techniques is the MLR (Multiple Linear Regression) (Cramer *et al*., 1988) method which quantifies the activity level of a descriptor array X as: $Activity(X) = c + \sum_{i=1}^{n} \sigma_i \cdot X[i]$ where c is a constant. If $Activity(X) \geq t$ for a (user specified) threshold value t then it is likely that the molecule is active with respect to the bioactivity of interest. Notice that the MLR classifier is described by a planar separator in the multi-dimensional descriptor array space; those points on one side of the separator are classified as active and those on the other side are classified as inactive. There are many different optimization criteria for determining the separator plane, i.e. the coefficients σ_i. The most widely used one (which we used in our experiments) is the partial least squares criteria (Geladi and Kowalski, 1986), which suggests to minimize the sum of the squares of differences between actual and predicted activity levels of the compounds in a training set. The separator plane which satisfies this criteria is NP-hard to compute deterministically but can be approximated through genetic algorithms, local search heuristics, etc.

Another popular statistical classification method is Linear Discriminant Analysis (LDA) (Livingston, 1995). Given a set of descriptor arrays, LDA computes a linear projection of the descriptor array space into a Euclidean space with 2 or 3 dimensions (i.e. each descriptor array is mapped to a point in the 2/3-D Euclidean space). The projection aims to maximize the ratio of between-class variance and within-class variance. The projection of descriptor arrays to points in the Euclidean space is followed by the computation of a line/plane which best separates the active and inactive compounds, i.e. maximizes the accuracy of the classifier. For a given query compound with unknown activity, its class is then simply determined by checking to which subspace its projection falls into; clearly this can be performed very fast.

It is also possible to perform compound classification via well known machine-learning techniques such as SVM (Support Vectors Machines) (Zernov *et al*., 2003) and, more commonly, ANN (Artificial Neural Networks) (Zupan and Gasteiger, 1999).

All these QSAR techniques (i.e. compound classifiers) have their own advantages and drawbacks. Statistical techniques such as LDA and MLR typically produce lower accuracy compared to the machine-learning approaches. On the other hand ANN only returns a binary value for the bioactivity (YES or NO) and provides no insight into the level of the bioactivity or the importance of the descriptors with respect to the bioactivity. It also does not provide a way of probing/similarity search, and can be somewhat slow.

Our contributions to compound classification

In this paper we focus on the k-nearest neighbor (k-nn) classification. which deduces the level of the bioactivity of a query molecule based on the number (and the bioactivity levels) of active elements among its k-nn with respect to a distance measure of choice. Although k-nn classification is a well known data mining method, it was not considered for small molecule classification until recently (Zheng and Tropsha., 2000; Itskowitz and Tropsha., 2005). The few known applications of k-nn method to compound classification aim to select the most relevant set of chemical descriptors to reduce the

size of the descriptor arrays used. The compounds are then compared under the standard (unweighted) L_1 or L_2 distance.

In this paper we introduce use of the (more general) weighted Minkowski distance of order 1, namely wL_1. For each bioactivity of interest, we determine real valued weights w_i of the wL_1 distance so as to maximize the discrimination between active and inactive compounds in a training set. (Thus, earlier applications of k-nn to compound classification can be seen as limited versions of our approach where the weights w_i are set to either 0 or 1.) We compute the *optimal* values for weights w_i via a linear optimization procedure.

Our experiments show that our k-nn classifier with respect to wL_1 distance provides better accuracy than the LDA and MLR, sometimes significantly so. Note that, as per LDA and MLR, our classifier is also based on a projection of molecules to a metric space. As per MLR (and in contrast to LDA) the number of dimensions in the projection space is equal to the number of descriptors. However, unlike MLR and LDA, our classifier is not described by a simple planar cut on the projection space but by a complex surface defined by the combination of surfaces in the form of *balls* with specific data elements in their center. Although our classifier uses more complex surfaces (which results in higher accuracy) we can still perform fast classification, thanks to the efficient data structures we develop for nearest neighbor—see below. Our method is comparable to the ANN classifier in terms of accuracy. Yet it is superior to the ANN classifier in the sense that it determines the level of bioactivity (rather than giving a simple YES or NO answer) as per the MLR based solutions. It turns out that our classifier is also faster than the ANN classifier—this we achieve through an efficient data structure we develop for efficient similarity search as described below.

Similarity search among small molecules

Efficient data structures for performing nearest neighbor search in high dimensional metric spaces usually exploit the triangle property satisfied by the distance measure. The primary example of these *distance based* proximity search data structures is the Vantage Point (VP) Trees (Uhlmann, 1991). In a VP tree, efficient similarity search in a large data set is achieved through iterative pruning. Among the data elements, the VP Tree randomly picks a Vantage Point V and partitions the data set into two equal size subsets according to their proximity to V. Those which are *close* to V form the *inner partition* and those which are *far* form the *outer partition*. The two subsets are further partitioned via the iterative application of the above procedure until each subset includes a single data element.

When performing a similarity search, the query element X is first compared to the Vantage Point of the entire set. If X is sufficiently close to V the search is performed in the its *inner partition*. If X is sufficiently far from V the search is performed in the *outer partition*. It is possible the X is neither too close nor too far; in this situation the search is performed simultaneously in both partitions implying that no pruning has been achieved.

A modification to traditional VP trees, which we call Space Covering VP Trees (or SCVP trees) was described by Sahinalp *et al*. (Sahinalp *et al*., 2003) to avoid situations in which pruning is not achieved. At each level of the SCVP tree there are multiple vantage points which are chosen in a way that the union of the inner partitions of these vantage points cover the entire data set. In other words, each data element is included in at least one of the inner partitions of a vantage point. Thus a SCVP tree has multiple branches at each internal node, each representing a vantage point and its inner partition. No branch exists for representing an outer partition. If a query element is not close to any of the vantage points at a given level, it is deduced that there are no similar items to it in the data set.

The SCVP trees introduce some redundancy in the representation of the data elements: clearly each data element may be included in more than one inner partition and thus need to be represented in more than one subtree. Thus the memory requirements of the SCVP tree can be fairly large. In case the full SCVP Tree requires more memory than available, some of the lower levels could be cut out—after which linear search needs to be employed.

Our contributions to similarity search among small molecules

In the original SCVP tree construction, the vantage points in each level are chosen randomly until all search space is covered (Sahinalp *et al*., 2003). Clearly, it is desirable to minimize the number of vantage points that cover the search space. With fewer vantage points picked at each level, a better space utilization can be achieved, implying that more levels of the tree can be fitted in the available memory.

We first prove that the problem of minimizing the number of vantage points at each level is an NP-hard problem. However, we show how to approximate the minimum number of vantage points and thus obtain the optimum allocation of available memory through a simple polynomial time algorithm. The resulting data structure, which we call the deterministic multiple vantage point tree (DMVP tree), when built in full, is guaranteed to have $O(\log \ell)$ levels, where ℓ is the size of the data set. If the maximum number of children of an internal node at level i is c_i, the query time guaranteed by our data structure is $O(\sum_{i=1}^{\log \ell} c_i$. Because c_i is typically a small constant, the query time is only $O(\log \ell)$, a significant improvement over linear/brute force search.

Due to redundant representation of data items, the memory usage of the DMVP tree can be super-polynomial. In case the full version of the DMVP tree requires more memory than available, lower levels of the DMVP trees could be cut out. In this case, when the search routine reaches the final level built, the pruning in the respective subspace can be achieved by linear search. We also show how to obtain the optimum cut so as to minimize the expected query performance.

Our data structure is not only interesting for classification purposes; similarity search among small molecules under various notions of similarity is of independent interest. To the best of our knowledge, this is the first application of an efficient similarity search data structure to small molecule data collections. In particular, all known k-nn classifiers employ brute force search, which is not scalable with the growth in the size of compound databases (e.g. PubChem).

We demonstrate that the DMVP tree performs very well in practice, achieving fast classification and similarity search. We compare the performance of our data structure against brute force search in terms of the number of comparisons between descriptor arrays that we need to perform under the weighted

Minkowski distance. We also demonstrate how well our classifier performs against available alternatives in terms of running time.

2 DISTANCE MEASURES FOR SMALL MOLECULES AND DISTANCE BASED CLASSIFICATION

Given a chemical compound s, its descriptor array S is defined to be an n dimensional vector in which each dimension i, denoted by $S[i]$, is a real value corresponding to the descriptor associated with dimension i. For a given bioactivity, it is of significant interest to come up with a distance measure $D(S, R)$ between pairs of descriptor arrays S and R that correspond to the similarity in the bioactivity levels of the corresponding compounds s and r: if the bioactivity levels are similar, the distance must be small and vice versa. Such a distance measure could be very useful in the classification of *new* chemical compounds in terms of the bioactivity of interest: the bioactivity level of the new compound is likely to be identical to the bioactivity level of the set of compounds that have the smallest distance to the new compound.

A distance measure D forms a metric if the following conditions are satisfied. (i) $D(S, S) = 0$ for all S and $D(S, R) \geq 0$ for all S and R (non-negativity). (ii) $D(S, R) = D(R, S)$ (symmetry). (iii) $D(S, R) \leq D(S, Q) + D(Q, R)$ (triangle inequality). Metric distance of interest include the Hamming distance, Euclidean distance and the Tanimoto distance. Metric distances are of particular interest due to the availability of efficient data structures they admit for fast similarity search.

The commonly used QSAR approach estimates the level of bioactivity of a compound via a linear combination of its descriptors each of which correspond to a specific dimension of its descriptor array. In *distance based* compound classification, it is thus natural to consider a distance between two descriptor arrays which is a linear combination of the differences in each one of the dimensions. More specifically one can define $D(S,R) = \sum_{i=1}^{n} w_i \cdot |\, S[i] - R[i] \,|$ where w_i, the weight of the dimension i is a real value in the range [0,1]. It is easy to show that this distance, which is usually called the weighted Minkowski distance of order 1 forms a metric.

In this paper we focus on classification of biomolecules according to five specific bioactivities: (i) being an antibiotic, (ii) being a bacterial metabolite, (iii) being a human metabolite, (iv) being a drug, and (v) being drug-like. The biomolecular data sets available usually do not specify the level of bioactivity of interest but rather provide whether a compound is active or inactive. Thus we only perform a binary classification of compounds for each bioactivity, although our methods are general to provide a real valued level of bioactivity.

Our classification method for a given bioactivity first computes a distance measure for a training data set which *separates* the subset of active compounds from those that are inactive. Given a training set of descriptor arrays $T = \{T_1, T_2, \ldots, T_\ell\}$ (each of which belonging to a compound) we determine the distance measure D, more specifically compute the associated weights w_i, through a combinatorial optimization approach.

Given the training set T, let $T^A = \{T_1^A, T_2^A, \ldots, T_m^A\}$ denote its subset of active compounds and $T^I = \{T_1^I, T_2^I, \ldots, T_{\ell-m}^I\}$ denote its subset of inactive compounds. Clearly $T = T^I \cup T^A$.

We obtain a linear program for determining each w_i as follows. The objective function of the linear program which is to be minimized is

$$f(T) = \Big(\sum_{h=1}^{m} \sum_{j=1}^{m} \sum_{i=1}^{n} w_i \cdot |\, T_h^A[i] - T_j^A[i] \,| \Big) / m^2 \tag{1}$$

$$+ \Big(\sum_{h=1}^{l-m} \sum_{j=1}^{l-n} \sum_{i=1}^{n} w_i \cdot |\, T_h^I[i] - T_j^I[i] \,| \Big) / (l-m)^2 \tag{2}$$

$$- \Big(\sum_{h=1}^{m} \sum_{j=1}^{l-m} \sum_{i=1}^{n} w_i \cdot |\, T_h^A[i] - T_j^I[i] \,| \Big) / (m \cdot (l-m)) \tag{3}$$

subject to the following conditions

$$\forall T_h^A \in T^A \Big(\sum_{j=1}^{m} \sum_{i=1}^{n} w_i \cdot |\, T_h^A[i] - T_j^A[i] \,| \Big) / m^2$$

$$\leq \Big(\sum_{j=1}^{\ell-m} \sum_{i=1}^{n} w_i \cdot |\, T_h^A[i] - T_j^I[i] \,| \Big) / (m \cdot (l-m)) \tag{4}$$

$$\forall i \; 0 \leq w_i \leq 1 \quad \& \quad \sum_{i=1}^{n} w_i \leq C \tag{5}$$

where C is a user defined constant.

The objective function $f(T)$ has three components: Component (1) is the average distance among active compounds and component (2) is the average distance among the inactive compounds; their sum provides the *within-class* average distance. Component (3), on the other hand, is the average distance between an active compound and an inactive one; thus it stands for the *between-class* average distance. As a result our linear programming formulation aims to maximize the difference between the average between-class distance and the average within-class distance. The distance measure obtained will *separate* the typical active compound from the typical inactive compound, while *clustering* all active compounds and all inactive compounds as much as possible.

There are three types of constraints on the weights w_i in our linear programming formulation. Constraint (4) ensures that the average distance among active compounds is no more than the average distance between active and inactive compounds.[2] Constraints (5) impose bounds on the values of weights w_i and their sum.[3]

A note on the performance. We used CPLEX, an open-source linear programming solver for computing the distance measure for a given bioactivity. Because the number of constraints is proportional to the number of active compounds, which is no more than 1500 for the bioactivities we considered, the running time for computing all

[2]A more stringent set of constraints can be imposed on active compounds such that the distance between a given active compound T_h^A and any other active compound is no more than the distance between T_h^A and any inactive compound. Such a set of constraints can, in principle, can separate active and inactive compounds into tighter clusters. Unfortunately, the number such constraints, $m^2 \cdot (\ell - m)$, turns out to be impractical, even for the most advanced linear program solvers.

[3]The number of descriptors related to a specific bioactivity is usually no more than a few, thus it is desirable to simplify the distance measure by limiting the number of non-zero weights. The final constraint aims to achieve this by imposing an upper bound on the sum of the weights. Although this constraint does not guarantee to upper bound the number of non-zero weights, in practice, the number of non-zero weights obtained are no more than $2C$.

distance measures of interest was quite reasonable, no more than 2 minutes on a standard 3.2Ghz Intel Pentium D Workstation.

k-nearest neighbor classification of biomolecules

A distance measure defined as above can be used for the classification of compounds with unknown levels of bioactivity as the bioactivity level of a compound is likely to be similar to the bioactivity levels of compounds within its close proximity. Our *k-nn classifier* estimates the (binary) bioactivity of a given compound by (1) either taking the majority of the bioactivities of its k-nearest compounds w.r.t. the distance measure or by (2) checking whether sum of the binary bioactivity levels of the k-nearest neighbors normalized by their distances to the compound is above a threshold value. Under each approach, it is possible to select the value of k which maximizes the accuracy of the estimator, i.e. the ratio of the sum of true positives and true negatives to the size of the training data set.

Once the method of classification is determined it is desirable to construct an efficient data structure for performing k-nn search. In the remainder of the paper we first discuss how well our approach compares with other popular methods for compound classification. Then we focus on how we construct an efficient k-nn search data structure for the distance measure we construct and provide some experimental results.

3 EFFICIENT DATA STRUCTURES FOR *K*-NN SEARCH

Typical similarity search methods for large collections of data elements usually perform iterative partitioning of the data set into smaller subsets so as to perform efficient querying by pruning—which is achieved at each iteration by checking out to which partition the query falls into (Uhlmann, 1991; Yianilos, 1993). The pruning strategy can be made particularly effective on data collections where similarity is measured with respect to a metric distance. The partitions in such a metric space are usually achieved with respect to simply defined planar cuts; given a query element, it is quite simple to check to which side of the planar cut it falls into.

Given a set of data elements $X = \{X_1, \ldots, X_\ell\}$ in a metric space with distance D, similarity search for a query element Y can be posed in two flavors. (1) Range query: retrieve all items whose distance to Y is at most some user defined R. (2) k-nn query: retrieve the $k \geq 1$ items whose distances to Y are as small as possible.

One particularly efficient similarity search tool for performing range queries is the Vantage Point (VP) trees (Uhlmann, 1991; Yianilos, 1993). Traditionally, a vantage point tree is defined as a binary tree that recursively partitions a data set into two equal size subsets according to a randomly selected vantage point X_v as follows. Let M is the median distance among the distances of the data elements to X_v. The *inner partition* consists of the elements Y such that $D(X_v, Y) < M$ and the *outer partition* consists of the elements Z such that $D(X_v, Z) \geq M$.

For a given query element Y, the set of data elements X_i for which $D(Y, X_i) \leq R$ for the search radius R can be computed as follows. Let X_v be the vantage point chosen for the entire data set and let M be the median distance among the distances of the data elements to X_v. If $D(X_v, Y) + R \geq M$ then recursively search the *outer partition*. If $D(X_v, Y) - R < M$ then recursively search the *inner partition*. If both conditions are satisfied then both partitions must be searched. The correctness of the search routine follows from the triangle inequality.

A natural extension to the traditional vantage point trees is what we call the *Space Covering VP trees* (SCVP Trees) first described by Sahinalp *et al*. (Sahinalp *et al*., 2003). At each level of the SCVP trees, multiple vantage points are chosen so as to increase the chance of inclusion of the query region in one of the inner partition of the vantage points. The original SCVP trees chose the vantage points at each level randomly. Although this approach can perform quite well for certain data collections, it can also result in poor space utilization.

Clearly it is desirable to *cover* the entire data collection by the fewest number of (inner partitions of) vantage points. However, the problem of minimizing the number of vantage points for this purpose turns out to be an NP-hard problem under all distance measures of interest (i.e. weighted Minkowski distance of any order p, wL_p); this is proven below. Nevertheless it is possible to approximate the minimum number of vantage points in any metric space through a simple polynomial time algorithm as we show later. As a result we obtain a data structure that deterministically picks the vantage points (whose inner partitions cover the entire data set) which results in almost optimal redundancy; we call this data structure *Deterministic Multiple Vantage Point* tree (DMVP tree).

We start with showing that the optimal vantage point selection problem, which we call OVPS problem, is NP-hard for any weighted Minkowski distance of order p, namely wL_p.

THEOREM 1. *OVPS problem under the weighted Minkowski distance of any order p is NP-hard.*

PROOF. We establish the NP-hardness of the OVPS problem under L_p through a reduction from the Dominating Set Problem which is known to be NP-hard. The decision version of the Dominating Set problem is as follows: Given a graph $G(V, E)$ and an integer k decide whether there exists a subset V' of vertices V such that every vertex in $V - V'$ has a neighbor in V'. The decision version of the OVPS problem in L_p is as follows: Given a set S of points in L_p, a radius r, and an integer k, decide whether there exists k (vantage) points such that the distance between each point in the set and at least one of the k points is less than r.

From an instance of the Dominating Set problem we first construct a $|V|$ dimensional space S where each vertex V_i is mapped to a point X_i in S as follows.

$$X_i[j] = \begin{cases} 1 & \text{if } i = j \\ -\epsilon & \text{if } (V_i, V_j) \notin E \\ 0 & \text{if } (V_i, V_j) \in E \end{cases}$$

One can calculate upper and lower bounds for the L_p distance between two vectors X_i and X_h as follows.

$$L_p(X_i, X_h)^p = \sum_{j=1}^{|V|} w_i \cdot |X_i[j] - X_h[j]|^p.$$

$$= \begin{cases} a \geq 2(1+\epsilon)^p & \text{if } (V_i, V_h) \notin E \\ b \leq 2 + \epsilon^p(|V| - 2) & \text{if } (V_i, V_h) \in E \end{cases}$$

If for a given p one picks ϵ such that

$$\epsilon < \frac{2p}{(|V| - 2)}$$

then

$$\binom{p}{p-1}\epsilon^{p-1} > \epsilon^p \frac{(|V| - 2)}{2}$$

which implies that

$$1 + \binom{p}{1}\epsilon + \ldots + \binom{p}{p-1}\epsilon^{p-1} + \binom{p}{p}\epsilon^p > 1 + \epsilon^p \frac{(|V| - 2)}{2}$$

and thus

$$2(1 + \epsilon)^p > 2 + \epsilon^p(|V| - 2)$$

which implies that

$$a > b.$$

In other words, b, the distance between any two vectors whose corresponding vertices are connected (by an edge) is less than a, the distance between any two vertices which are not connected. We now simply pick r so that $a > r > b$.

We now show that G has a dominating set of size k if and only if there exists k vantage points for which the distance between each point in the data set S and at least one of the vantage points is at most r. Given G, and a dominating set D of size k, we show that the k points in S that correspond to the k vertices in D, *cover* the entire set S. For any vertex $V_i \notin D$, there must exist a neighboring vertex $V_h \in D$. But if V_i and V_h are neighbors then by the above argument $L_p(X_i, X_h) < r$, i.e. X_i is in the radius-r-neighborhood of the vantage point X_h.

Given S, and k vantage points whose radius-r-neighborhoods cover all points in S, we show that the k vertices in G that correspond to the k vantage points form a dominating set. For any point X_i which is not a vantage point, there must exist a vantage point X_h s.t. $wL_o(X_i, X_h) < r$. But this implies that V_i and V_h must be neighbors in G, i.e. V_i must have a neighbor which is in the dominating set.

The generalization of the proof to wL_p is not difficult and is not given here.

COROLLARY 2. *OVPS problem under Tanimoto distance is NP-hard.*

PROOF. The Tanimoto distance is no more than L_1 on binary vectors normalized by the number of dimensions (which is a constant).

An $O(\log \ell)$ approximation to the optimal vantage point selection

The variant of the OVPS problem for which we establish NP-hardness assumes a fixed radius r for each neighborhood around a vantage point. One can think of two natural variants of the OVPS problem: (1) each neighborhood includes a fixed number of points (e.g. $\ell/2$ points as per the original VP Tree construction), (2) each neighborhood has at least ℓ/k and at most ℓ/k' points for some $k \geq k'$. It is not difficult to show that these variants are NP-hard as well.

In the remainder of the paper we focus on variant (2) of the OVPS problem and describe a polynomial time $O(\log \ell)$ approximation algorithm for solving it. Such a solution will also imply an $O(\log \ell)$ approximation algorithm for variant (1) by setting $k = k'$. The approximation algorithm is achieved by reducing the OVPS problem to the weighted set cover problem as follows.

Consider each point X_i in S. We construct the following ℓ sets for X_i named $X_i^1, X_i^2, \ldots, X_i^\ell$. X_i^1 consists of only X_i. X_i^2 consists of X_i and its nearest neighbor. In general, X_i^j consists of X_i and its $j-1$ nearest neighbors. Let the cost of X_i^j be j.

Now given sets X_i^j, for all $1 \leq i \leq \ell$ and $k \leq j \leq k'$, each with cost j, if we can compute the minimum cost collection of sets such that each $X_h \in S$ is in at least one such set, we would get a solution to the variant (2) of the OVPS problem. This problem is equivalent to the weighted set cover problem for which a simple greedy algorithm provides an $O(\log \ell)$ approximation (e.g. (Chvatal, 1979)). The greedy algorithm works iteratively: each iteration simply picks a set where the cost-per-uncovered-element is minimum possible. The algorithm terminates when all elements are covered.

Optimal fitting of the multiple vantage point tree in the memory

Although the deterministic multiple vantage point tree improves the memory usage of the randomized space covering vantage point tree, it is still possible that the tree may not fit in the main memory. If this is indeed the case, we try to place a connected subtree (which includes the root) to the memory. The search again is performed starting with the root. When an internal node whose children are not represented in the memory is reached, the search is done in a brute force manner on the set of points represented by that node.

Clearly it is of interest to obtain the *best* subtree for optimizing the query performance of the data structure. For that we use the following $0 - 1$ programming formulation.

Given a Multiple Vantage Point tree T and a node i, let S_i be the number of points in the neighborhood represented by i. During a search, when a node j is reached, its children i, $i + 1$, ... are considered for further search in linear order; i.e. we first check whether the query fits in the neighborhood of i, then we check $i + 1$ and so on until a suitable vantage point $i + h$ is found. Let S'_{i+h} be the number of points in the neighborhood represented by node $i + h$ which are not in the neighborhoods represented by $i, i + 1, \ldots, i + h - 1$.

Our $0 - 1$ programming formulation sets the *probability* that node $i+h$ is reached during a search to S'_{i+h}/ℓ. If the children of the node $i + h$ are not placed in the memory, i.e. if node $i + h$ is on the *cut-set*, the time needed for performing a search on the neighborhood represented by this node is S_{i+h}. Thus the expected contribution of node $i + h$ to the query time is $S_{i+h} \cdot S'_{i+h}/\ell$.

Let b_i be a binary variable, which takes the value 1 if vertex i is in the cut-set and is 0 otherwise. Our goal is to minimize the expected running time of the brute-force search performed for each query; i.e. our objective function is $f(T) = \sum_{\forall i} b_i S_i S'_i$ subject to the following constraints.

For any pair of consecutive sibling nodes i and $i+1$, we must have $b_i = b_{i+1}$.

We should not exceed the memory M dedicated to the cut-set; thus $\sum_{\forall i} b_i S_i \leq M$. Finally, at least one node in every path from the

Table 1. Binary classification of the bioactivities of the test set according to four classification methods: *k*-nn, LDA, MLR, ANN

Model		T_P	T_N	F_P	F_N	SPEC	SENS	ACCUR	PPV	NPV
Antibacterial Model, C=∞	Train	269	2610	69	95	0.97	0.74	0.95	0.8	0.96
	Test	117	1119	28	39	0.98	0.75	0.95	0.81	0.97
Antibacterial Model, C=10	Train	224	2538	141	140	0.95	0.62	0.91	0.61	0.95
	Test	92	1085	62	64	0.95	0.59	0.90	0.60	0.94
Antibacterial Model, C=3	Train	201	2526	153	163	0.94	0.55	0.90	0.57	0.94
	Test	75	1074	73	81	0.94	0.48	0.88	0.51	0.93
Antibacterial Model, LDA	Train	364	0	2679	0	0.00	1.00	0.12	0.12	-
	Test	156	0	1147	0	0.00	1.00	0.12	0.12	-
Antibacterial Model, MLR	Train	194	564	2115	170	0.21	0.53	0.25	0.08	0.77
	Test	61	1129	18	95	0.98	0.39	0.91	0.77	0.92
Antibacterial Model, ANN	Train	294	2651	27	70	0.99	0.81	0.97	0.92	0.97
	Test	129	1132	16	27	0.99	0.83	0.97	0.89	0.98
Bacterial Metabolite Model, C=∞	Train	311	2537	112	83	0.96	0.79	0.94	0.74	0.97
	Test	135	1091	44	33	0.96	0.80	0.94	0.75	0.97
Bacterial Metabolite Model, C=10	Train	220	2436	213	174	0.92	0.56	0.87	0.51	0.93
	Test	98	1038	97	70	0.91	0.58	0.87	0.50	0.94
Bacterial Metabolite Model, C=3	Train	152	2376	273	242	0.90	0.39	0.83	0.36	0.90
	Test	80	1018	117	88	0.90	0.48	0.84	0.41	0.92
Bacterial Metabolite Model, LDA	Train	240	2587	62	154	0.98	0.61	0.93	0.79	0.94
	Test	90	1088	47	78	0.96	0.54	0.90	0.66	0.93
Bacterial Metabolite Model, MLR	Train	301	2525	124	93	0.95	0.76	0.93	0.71	0.96
	Test	119	1073	62	49	0.95	0.71	0.91	0.66	0.96
Bacterial Metabolite Model, ANN	Train	338	2597	52	55	0.98	0.86	0.96	0.87	0.98
	Test	159	1076	59	10	0.95	0.94	0.95	0.73	0.99
Drug Model, C=∞	Train	474	2158	214	197	0.91	0.71	0.86	0.69	0.92
	Test	204	928	88	83	0.91	0.71	0.87	0.70	0.92
Drug Model, C=10	Train	349	2072	300	322	0.87	0.52	0.80	0.54	0.87
	Test	151	861	155	136	0.85	0.53	0.78	0.49	0.86
Drug Model, C=3	Train	305	2026	346	366	0.85	0.45	0.77	0.47	0.85
	Test	126	846	170	161	0.83	0.44	0.75	0.43	0.84
Drug Model, LDA	Train	0	2372	0	671	1.00	0.00	0.78	-	0.78
	Test	0	1014	2	287	0.99	0.00	0.78	0.00	0.78
Drug Model, MLR	Train	279	2234	138	392	0.94	0.42	0.83	0.67	0.85
	Test	109	951	65	178	0.94	0.38	0.81	0.63	0.84
Drug Model, ANN	Train	489	2178	194	182	0.92	0.73	0.88	0.72	0.92
	Test	177	978	39	110	0.96	0.62	0.89	0.82	0.90
Druglike Model, C=∞	Train	674	2043	158	168	.93	0.80	0.89	0.81	0.92
	Test	281	866	77	79	.92	0.78	0.88	0.78	0.92
Druglike Model, C=10	Train	560	1959	242	282	.89	0.67	0.83	0.70	0.87
	Test	239	842	101	121	.89	0.66	0.83	0.70	0.87
Druglike Model, C=3	Train	467	1813	388	375	.82	0.55	0.75	0.55	0.83
	Test	197	275	168	163	.82	0.55	0.75	0.54	0.83
Druglike Model, LDA	Train	683	1917	284	159	0.87	0.81	0.85	0.71	0.92
	Test	295	801	142	65	0.85	0.82	0.84	0.68	0.92
Druglike Model, MLR	Train	665	1951	250	177	0.89	0.79	0.86	0.73	0.92
	Test	282	812	131	78	0.86	0.78	0.84	0.68	0.91
Druglike Model, ANN	Train	734	2086	114	107	0.95	0.87	0.93	0.87	0.95
	Test	334	891	52	27	0.94	0.93	0.94	0.87	0.97
Human Metabolite Model, C=∞	Train	773	2270	0	0	1.00	1.00	1.00	1.00	1.00
	Test	331	972	0	0	1.00	1.00	1.00	1.00	1.00
Human Metabolite Model, C=10	Train	772	2266	4	1	0.99	0.99	0.99	0.99	0.99
	Test	330	972	0	1	1.00	0.99	0.99	1.00	0.99
Human Metabolite Model, C=3	Train	772	2270	0	1	1.00	0.99	0.99	1.00	0.99
	Test	330	972	0	1	1.00	0.99	0.99	1.00	0.99
Human Metabolite Model, LDA	Train	773	2270	0	0	1.00	1.00	1.00	1.00	1.00
	Test	331	972	0	0	1.00	1.00	1.00	1.00	1.00
Human Metabolite Model, MLR	Train	773	2270	0	0	1.00	1.00	1.00	1.00	1.00
	Test	331	972	0	0	1.00	1.00	1.00	1.00	1.00
Human Metabolite Model, ANN	Train	773	2270	-0	0	1.00	1.00	1.00	1.00	1.00
	Test	331	972	0	0	1.00	1.00	1.00	1.00	1.00

root to a leaf in T must include one vertex in the cut-set. Thus for any such path P we have $\sum_{i \in P} b_i = 1$.

A $0-1$ assignment to b_i's that minimize the objective function will minimize the expected query time while fitting the data structure in the main memory.

4 PRELIMINARY EXPERIMENTS

In this section we aim to provide some insight into the comparative performance of our k-nn classifier, both in terms of accuracy and efficiency. We applied our classifier to five types of bioactivities: (i) being antibiotic, (ii) being a bacterial metabolite, (iii) being a human metabolite, (iv) being a drug, and (v) being drug-like.

The first data set we used is the complete small molecule collection from (Cherkasov, 2005), which includes 520 antibiotics, 562 bacterial metabolites, 958 drugs, 1202 drug-like compounds, and an additional 1104 human metabolites. The total number of the compounds in the data set is 4346. Each compound in the dataset is represented with a descriptor array of 62 dimensions, which is a combination of 30 inductive QSAR descriptors (Cherkasov, 2005) and 32 physicochemical properties such as molecular weight, number of specific atoms (O, N, S), acidity, density, etc. This data set was used for testing the classification quality of our approach. A second data set which enriches the first data set by the addition of 20000 additional drug like compounds was later used for testing the running time of our approach. For each bioactivity, a wL_1 distance is determined to establish a *model* for compound classification w.r.t. this bioactivity using our k-nn method. Note that the descriptors of each compound are normalized according to the observed maximum and minimum values in the data set in order to remove the bias to parameters with larger values.

The comparative results of the four classification methods, namely k-nn, LDA,MLR and ANN are provided in Table 1. For each bioactivity, we provide the sensitivity,specificity and accuracy obtained by each classifier. We demonstrate the performance of our k-nn classifier only for $k = 1$; i.e. given a query compound, our classifier returns the bioactivity of its nearest neighbor in the training data set. We constructed the wL_1 measure for three different values of C-the upper bound on the sum of weights, i.e., $\sum_{i=1}^{n} w_i \leq C$. Setting $C = \infty$ removes the restriction on the sum of weights and thus computes the wL_1 distance that achieves the best classification. We also set C to 3 and 10 to restrict the number of non-zero weights, with the aim of focusing only on the C most relevant descriptors to the bioactivity of interest. As the resulting non-zero weights turned out to be equal to or very close to 1, these two classifiers are quite similar to those described in recent papers (e.g. (Zheng and Tropsha., 2000; Itskowitz and Tropsha., 2005)) that focus on determining the most relevant descriptors for modeling a bioactivity of interest.

We used MOE(Molecular Operating Environment) PLS module for MLR classification and SNNS (Stuttgart Neural Network Simulator) with default parameters (52 nodes and 420 connection network) for ANN classification.

LDA classification is performed through the use of standard C libraries for matrix operations.

For each bioactivity, a *training data set* comprising of 70 percent of both the active and the inactive compounds are formed via random selection. The remaining compounds are used as the *test data set*. Each training data set is used for building the four classifiers corresponding to the related bioactivity and the test data is used for the evaluating their performance.

For each bioactivity/classifier pair we report the following test results: The number of true positives (T_P), the number of true negatives (T_N), the number of false positives (F_P), the number of false negatives (F_N), sensitivity (T_P/(T_P+F_N)), specificity (T_N/(T_N+F_P)), accuracy ((T_N+T_P)/(T_P+T_N+F_P+F_N)), positive predictive value(T_P/(T_P+F_P)), negative predictive value (T_N/(T_N+F_N)).

Our similarity search data structure for computing the nearest neighbor of the query compound is quite efficient, especially when compared to brute force search. We tested our data structure under the wL_1 distance computed for each of the five bioactivities, on both of the data sets. The crucial parameter that determines the performance of our data structure is the pruning it achieves for any given query compound. Thus we determined the percentage of compounds pruned in the second training data set (the first training data set enriched with 20000 drug like compounds), averaged over all compounds in the test data set. On a 32GB Sun Fire V40Z server (with 2.4 Ghz AMD 64bit Opteron processor) the respective pruning ratios are as follows. We achieved (i) 84.4% pruning for being antibiotic, (ii) 84.5% pruning for being bacterial metabolite, (iii) 86.1% pruning for being human metabolite, (iv) 81.7% pruning for being drug, and (v) 81% pruning for being drug-like. This is significant improvement over brute force search.

As a result our k-nn classifier turns out to be very fast. On the first data set, the running time of our k-nn classifier averaged over all 4346 compounds (training+test data sets) and all five bioactivities is 0.3 milliseconds on the above server. In contrast the ANN classifier requires 39.7 milliseconds on the same data set. On the second data set (which simply has additional 20000 compounds in the data structure) the running time of our k-nn classifier increases only to 1.3 milliseconds (again averaged over the 4346 compounds from the first data set and five bioactivities), still 30 times better than the ANN trained over a much smaller set.

5 CONCLUSION

We have demonstrated that our k-nn classifier with respect to wL_1 distance obtains better accuracy than the LDA and MLR, sometimes significantly so. It is comparable to the ANN classifier in terms of accuracy and is superior in the sense that it is capable of determining a real valued level of bioactivity rather than giving a simple YES or NO answer. Our classifier is and it is faster, thanks to the DMVP tree data structure we develop for fast similarity search. Our DMVP tree data structure improves the existing vantage point tree data structures in multiple ways. It provides a deterministic selection of the optimal vantage points in each level as well as providing the optimal cut of the tree so as to fit it in the available memory. Our data structure can be applied to any metric distance including the wL_p distance for any p and the Tanimoto distance. It performs very well in practice, achieving fast similarity search and classification.

REFERENCES

Sahinalp,S.C., Tasan,M., Macker,J. and Ozsoyoglu,Z.M. (2003) Distance-Based Indexing for String Proximity Search. *Proc. IEEE Int. Conf. on Data Eng.*, **19**, 135–138.

Chen,X. and Reynolds,C.H. (2002) Performance of Similarity Measures in 2D Fragment-Based Similarity Searching: Comparison of Structural Descriptors and Similarity Coefficients. *J. Chem. Inf. & Comp. Sci.*, **42**, 1407–1414.

Chvatal,V. (1979) A Greedy Heuristic for the Set Covering Problem. *Math. of Operations Research*, **4**, 233–235.

Cherkasov,A. (2005) Inductive Descriptors. 10 Successful Years in QSAR. *Curr. Computer-Aided Drug Des.*, **1**, 21–42.

Maggiora,G.M. and Johnson,M.A. (1990) Concepts and Applications of Molecular Similarity. *Wiley,* New York.

Uhlmann,J.K. (1991) Satisfying general proximity/similarity queries with metric trees. *Inf. Proc. Lett.*, **4**, 175–179.

Yianilos,P. N. (1993) Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces. *Proc. ACM-SIAM Symp. on Discr. Alg.*, **1**, 311–321.

Brown,R.D. (1997) Descriptors for Diversity Analysis. *Persp. Drug Discovery Des.*, **7/8**, 31–49.

Adamson,G.W., Cowell,J., Lynch,M.F., McLure,A.H.W., Town,W.G. and Yapp,A.M. (1973) Strategic Considerations in the Design of a Screening System for Substructure Searches of Chemical Structure Files. *J. Chem. Doc*, **13**, 153–157.

Zheng,W. and Tropsha,A. (2000) Novel Variable selection quantitative structure-property relationship approach based on the k-nearest neighbor principle. *J. Chem. Inf. & Comp. Sci.*, **40**, 185.

Itskowitz,P. and Tropsha,A. (2005) Kappa Nearest neighbors QSAR modeling as a variational problem: theory and applications. *J. Chem. Inf. Model.*, **45**(3), 777–85.

Good,A.C., So,S.S. and Richards,W.G. (1993) Structure-Activity relationships from Molecular similarity Matrices. *J. Medicinal Chemistry*, **36**, 433–438.

, MACCS II Manual, MDL Information Systems, Inc 14600 Catalina Street, San Leandro, CA 94577 USA.

Willett,P., Banard,J.M. and Downs,G.M. (1998) Chemical Similarity Searching. *J. Chem. Inf. & Comp.. Sci.*, **38**(6), 983–996.

Cramer,R.D., Bunce,J.D. and Patterson,D.E. (1988) Crossvalidation, Bootstrapping, and Partial Least Squares Compared with Multiple Regression in Conventional QSAR Studies. *Quant. Struct.-Act. Relat.*, **7**, 18–25.

Livingstone,D. J. (1995) Data analysis for chemists. Applications to QSAR and chemical product design. *Oxford Univ. Press* 239.

Zupan,J. and Gasteiger,J. (1999) Neural Networks in Chemistry and Drug Design, 2nd ed. *Wiley*, New York.

Geladi,P. and Kowalski,B.R. (1986) Partial Least-Squares Regression: A Tutorial. *Analytica Chimica Acta*, **185**, 1–17.

Zernov,V.V., Balakin,K.V., Ivaschenko,A.A., Savchuk,N.P. and Pletnev,I.V. (2003) Drug Discovery Using Support Vector Machines. The Case Studies of Drug-likeness, Agrochemical-likeness, and Enzyme Inhibition Predictions. *J. Chem. Inf. & Comp. Sci.*, **43**(6), 2048–2056.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e252–e259
doi:10.1093/bioinformatics/btl236

Rapid knot detection and application to protein structure prediction

Firas Khatib*, Matthew T. Weirauch and Carol A. Rohl**

Department of Biomolecular Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064

ABSTRACT

Motivation: Knots in polypeptide chains have been found in very few proteins, and consequently should be generally avoided in protein structure prediction methods. Most effective structure prediction methods do not model the protein folding process itself, but rather seek only to correctly obtain the final native state. Consequently, the mechanisms that prevent knots from occurring in native proteins are not relevant to the modeling process, and as a result, knots can occur with significantly higher frequency in protein models. Here we describe Knotfind, a simple algorithm for knot detection that is fast enough for structure prediction, where tens or hundreds of thousands of conformations may be sampled during the course of a prediction. We have used this algorithm to characterize knots in large populations of model structures generated for targets in CASP 5 and CASP 6 using the Rosetta homology-based modeling method.

Results: Analysis of CASP5 models suggested several possible avenues for introduction of knots into these models, and these insights were applied to structure prediction in CASP 6, resulting in a significant decrease in the proportion of knotted models generated. Additionally, using the knot detection algorithm on structures in the Protein Data Bank, a previously unreported deep trefoil knot was found in acetylornithine transcarbamylase.

Availability: The Knotfind algorithm is available in the Rosetta structure prediction program at http://www.rosettacommons.org

Contact: bort@soe.ucsc.edu

1 INTRODUCTION

In a formal topological sense, knots in protein chains cannot be defined because the protein backbone, disregarding disulfide bridges and other sources of backbone crosslinks, does not form a closed loop. Jane Richardson (1977) was the first to define a knotted protein chain as one which cannot be fully extended to a straight line if one were to grab the N- and C-terminus in each hand and pull. Few protein structures have been observed to contain knots in their backbones (Nureki *et al*., 2002), and in most cases where knots have been observed, they tend to be simple overhand knots near one terminus (Mansfield, 1994). These knots could in theory form by threading a short section of the polypeptide chain through a loop formed by another backbone section. Such knots disappear if a few residues are trimmed from the terminal ends (Taylor, 2000). Deep knots, in contrast, occur far from the protein chain termini and have been rarely observed.

Because knots in protein structures are rare, protein structure prediction methods should generally avoid introducing knots into the polypeptide backbone. Most structure prediction methods do not, however, check for knots. Additionally, few protein structure prediction methods model the kinetic protein folding process, so the entropic mechanisms that have been cited as explanations for the relative absence of knots in protein structures (Taylor, 2000) are not likely to prevent the introduction of knots in the modeling process. In fact, algorithms used for structure prediction do introduce knots in the polypeptide backbone, as demonstrated by predictions made for the Comparative Assessment of Methods for Structure Prediction (CASP) experiments (Moult *et al*., 1995). In the CASP 4 protein structure prediction experiment, one submitted model was assessed as being reasonably accurate in terms of atomic coordinates, but was also described by the CASP assessors as an ''impossible structure'' because it contained a trefoil knot (Tramontano *et al*., 2001). In the most recent CASP 6 experiment, the assessors reported that knotted models were still being submitted and that such knotted models submitted for comparative modeling targets were rejected out of hand without additional assessment (Tress *et al*., 2005a).

Knots in polypeptides can be difficult to detect by visual inspection alone, as evidenced by the fact that the assessors accepted some knotted CASP 6 models, presumably because it was not apparent that these models contained knots. Algorithms for automated knot detection have been reported (Taylor, 2000) but are too slow for general use in structure prediction, where tens or hundreds of thousands of conformations may need to be examined in the course of a single structure prediction. Here we present Knotfind, a rapid algorithm for knot detection, and report its application in the context of the Rosetta homology-based structure prediction method (Bradley *et al*., 2003; Rohl *et al*., 2004a). Additionally, the algorithm was applied to experimentally-determined protein structures in the Protein Data Bank (PDB; Berman *et al*., 2000) identifying a previously unreported deep trefoil knot.

2 METHODS

2.1 Knot-detection algorithm

The Knotfind algorithm considers only Cα atoms in a single protein chain and progressively 'eliminates' atoms from the Cα trace to simplify the chain. Triples of consecutive Cα atoms, i-1, i, i+1, are considered, ordered by

*To whom correspondence should be addressed.

**Current address: Rosetta Inpharmatics LLC, a wholly owned subsidiary of Merck & Co., Inc., 401 Terry Avenue N., Seattle, WA 98195

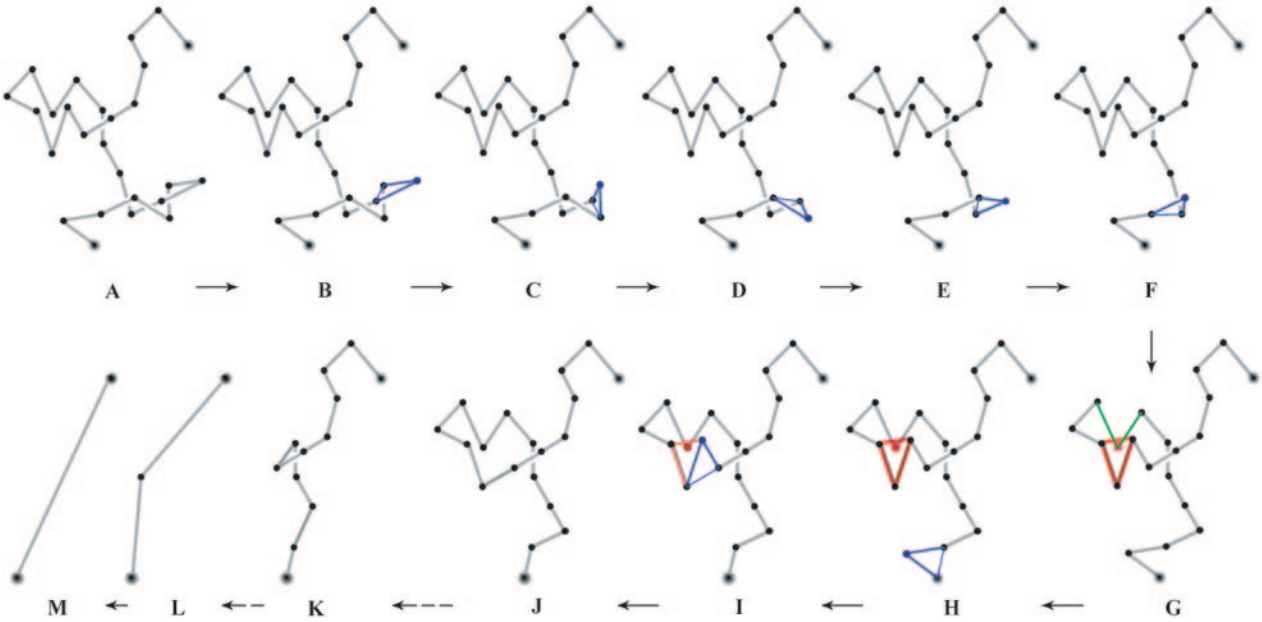


Fig. 1. Schematic illustration of the Knotfind algorithm. Starting from the initial Cα trace (trace A), atoms are progressively eliminated from the chain, effectively simplifying it to a straight line. For steps shown in traces B to F, the central Cα atom in the triple describing the most acute triangle (triangle shown in blue) is removed. In trace G, the most acute triangle cannot be simplified because two line segments (green) pass through this triangle; removing the central atom would effectively result in passing the red chain segment through the green segments. Since the red triple cannot be simplified, the triple forming the second most acute triangle (blue) is targeted for simplification in trace H. In trace I, the red triple still cannot be simplified, but the triple forming the next most acute triangle (blue) can be, yielding trace J. Trace K is obtained by seven additional atom removals and trace L by nine additional simplifications.

increasing Cartesian distance between atoms i-1 and i+1. For an individual triple, if no line segments connecting consecutive Cα atoms j, j+1 (for all $j < i$-1 and $j > i$+1) cross through the triangle defined by i-1, i, i+1, then Cα i is removed from the chain. If any line segment connecting two consecutive Cα atoms intersects the triangle, however, then no simplification of this triple is made and the algorithm proceeds to the triple with the next shortest i-1, i+1 distance. After any Cα is eliminated from the chain, the algorithm returns to the triple with the shortest i-1, i+1 distance. This procedure is repeated until the last triple in the distance list has been selected and simplified, if possible. When the algorithm terminates, if the only atoms left in the chain are the N- and C-terminal Cα atoms such that chain has been simplified to a straight line, the protein contains no knots (Figure 1). If, instead, the chain cannot be fully simplified to a single extended segment, the chain contains one or more knots and the remaining Cα atoms in the chain define the knotted region. In cases where a knot is detected, the algorithm is repeated using an alternate scheme to order the triples for simplification in which the area of the triangle defined by each i-1, i, i+1 triple is used in place of the i-1, i+1 interatomic distance to reduce false positives.

To determine if a line segment intersects a triangle, the algorithm first ensures that the plane containing the triangle and the line containing the line segment are not parallel, and then determines if both endpoints of the line segment lie on the same side of the plane. For segments that intersect the plane of the triangle, the algorithm determines if the intersection point lies within the triangle, relying on the fact that the sum of the internal angles of a point inside a triangle is 2π. Thus, any point lying outside the triangle will have smaller angle sums (http://astronomy.swin.edu.au/~pbourke/geometry/linefacet). An effective line width of 0.0003 is used in order to handle round off errors on arccosines in computing angle sums.

The Knotfind algorithm has been implemented in the Rosetta structure prediction program available at http://www.rosettacommons.org and in the Undertaker program (Karplus *et al*., 2005).

2.2 Protein structures

The PISCES server was used to identify 9,553 protein chains in the RCSB PDB as of February 12, 2006 with less than 90% sequence identity, with x-ray structures of resolution better than 3.0Å and no R-factor filtering ($R \leq 1.0$) (Wang *et al*., 2003). This list was supplemented with four protein chains that have previously been reported to be knotted (1dmxA, 1fugA, 1yveI, and 2btv), but which did not meet the resolution or sequence identity cutoffs, giving a total list of 9,557 chains that were examined using the Knotfind algorithm. Coordinate files were obtained from the RCSB PDB and ATOM records were compared to the sequence as defined in the SEQRES header to define regions of missing density. Missing density leading to a significant chain discontinuity (i.e. multiple residues not at a chain terminus) can make identification of a knot ambiguous because Cα atoms surrounding the missing density are artificially connected in a Cα trace. Consequently, structures with missing density that were reported by the algorithm to be knotted were visually inspected for confirmation to eliminate those that did not actually contain a knot. Among the 9,557 chains checked, seven knotted structures detected by Knotfind could be attributed to significant missing density: 1gkuB, 1jr1A, 1mqsA, 1o6lA, 1u2zA, 1yc0A, and 2bm0A.

2.3 Rosetta decoy sets

For predicted structures, models generated during the course of structure predictions made for CASP 5 and CASP 6 were utilized. Many structure prediction methods, including Rosetta, generate large numbers of possible model structures, referred to as 'decoys', from which a final best model is then selected. Decoy structures for CASP 5 and CASP 6 targets were generated using the Rosetta homology-based modeling method (Bradley *et al*., 2003; Rohl *et al*., 2004a) during the process of the CASP experiments. The CASP 5 decoy sets were generated by the Baker group (Group 2) and exclude decoy sets for any targets for which the de novo Rosetta prediction

method was used (Bradley *et al*., 2003). A total of 45,366 decoys were examined here. Decoy sets for individual targets included between 199 and 4,019 decoys. Decoy populations for CASP 6 were those generated during the course of predictions made by the Rohl group (Group 079), also using the Rosetta homology-based method. A total of 119,543 decoys were examined. Decoy sets for individual targets included between 883 and 11,934 decoys. In addition to manually generated Rosetta decoys, models generated by the automated Robetta server, which utilizes the Rosetta method, were also examined. Robetta predictions (Group 101) for CASP 6 targets were obtained from the Robetta server (Chivian *et al*., 2003; Kim *et al*., 2004) and are also available from the CASP 6 website (http://predictioncenter.org/casp6).

All Rosetta decoy sets used here, including models from the Robetta server, use the same basic Rosetta homology-based structure prediction method which has been described elsewhere (Bradley *et al*., 2003; Chivian *et al*., 2005). In brief, predictions begin from an alignment to a parent protein of known structure. Coordinates for aligned regions are taken directly from the parent structure and serve as a fixed template. Coordinates for structurally variable regions (SVRs), corresponding to both gaps in the alignment as well as regions of uncertain alignment, are constructed by assembling short fragments of known structure. These fragments are selected from the database of known protein structures based on similarity of sequence and predicted and known secondary structures. For short SVRs, geometric fit to the template is also considered. The selected fragments are combined using a Monte Carlo simulated annealing search by means of a knowledge-based potential function derived from the observed distributions of residues in known protein structure along with a gap penalty to ensure chain continuity in the final model. A more detailed description of the Rosetta approach and the potential function (Rohl *et al*., 2004b), and the SVR modeling method (Rohl *et al*., 2004a) are described in detail elsewhere. Differences between the CASP 5 and CASP 6 SVR modeling methods are described below.

For CASP 5 decoys, a library of possible conformations was selected via a database search for SVRs shorter than 17 residues. For each decoy, a random conformation for each short SVR was selected and then long SVRs were modeled by fragment assembly in the context of the template. For CASP 6 decoys, a library of possible conformations was generated for every SVR, regardless of length using a combination of database search and fragment assembly. For short SVR regions with 7 or fewer residues, conformations were selected directly from the database and used without further modification. For SVRs in the length range of 8-12 residues, conformations were assembled from 3-9 residue fragments in the context of the entire fixed template. For SVRs greater than 12 residues in length, a reduced template of four residues, two on each side of the SVR, was extracted, and the long SVR was modeled in the context of this reduced template by fragment assembly. For each SVR, regardless of length, 100-200 conformations were initially selected or generated and each of these library conformations was then checked using the Knotfind algorithm to eliminate those that resulted in knots when grafted onto the fixed template. Additionally, conformations with significant steric clashes with the template or large chain discontinuities were discarded. Complete models were then constructed by combining conformations from these libraries, using a Monte Carlo simulated annealing search to optimize the Rosetta centroid-based energy function.

2.4 Undertaker decoy sets

Undertaker decoys for CASP 6 targets were graciously provided by Kevin Karplus. A total of 2,373 decoys were examined. Decoy sets for individual targets included between 6 and 115 decoys. The Undertaker program combines fragment assembly and other methods with coordinate information extracted from alignments to a parent structure in order to generate models for proteins (Karplus *et al*., 2005). Rosetta and Undertaker are substantially different in terms of the optimization strategies and cost functions used, but share substantial similarity in their approach to conformation modification, which includes fragment assembly. Undertaker differs from the Rosetta-based strategy employed for construction of the Rosetta decoy sets used here in that regions modeled on the basis of homology to a parent of known structure are not treated as a fixed template in Undertaker, but instead are subject to conformational modifications.

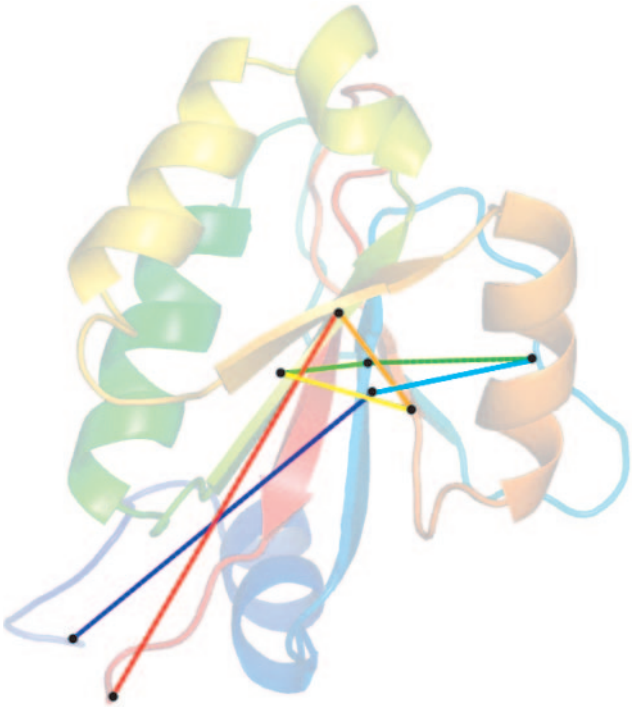


Fig. 2. The trapped state obtained for 1ogdA when simplifying triples in decreasing order of acuteness. Protein chain 1ogdA is shown as a ribbon and the final state resulting from the Knotfind algorithm, applied with triples ordered according to i-1, i+1 distance, is shown overlaid with the eight unsimplified Cα's indicated by black spheres. At this point, no triple can be simplified, yet this protein chain does not contain a knot. When triples are considered in order of their area by Knotfind, the chain completely simplifies.

3 RESULTS

3.1 Knotfind algorithm

The Knotfind algorithm attempts to simulate the process of pulling the protein chain from both ends in order to determine if the chain contains a knot. As described by Richardson's operational definition, an unknotted chain can be completely pulled into a fully extended conformation. In the presence of a knot, however, the chain cannot be fully extended without one segment of the chain being passed through another segment of the backbone. In the Knotfind algorithm, the chain pulling is modeled by progressively removing atoms from the chain. For each atom removal, all other segments of the backbone are checked to ensure that removal of an atom does not effectively cause one segment of the backbone to pass through another.

Simplifying the chain in a series of discrete steps allows the Knotfind algorithm to be fast, but leaves open the possibility that the chain trace can become trapped in a partially simplified state that does not contain a knot but cannot be further simplified according to

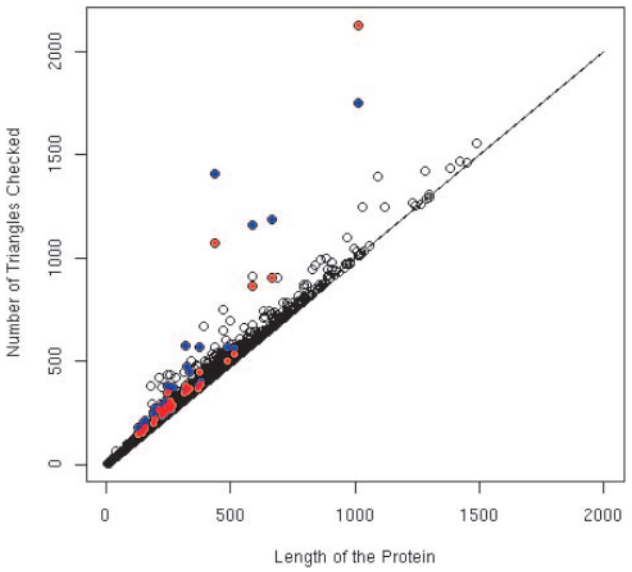


Fig. 3. The number of triples checked by Knotfind as a function of protein chain length for the 9553 protein chains taken from the PISCES server. Colored dots indicate protein chains reported as knotted by one of the triple ordering schemes used by Knotfind (blue: *i*-1, *i*+1 distance; red: triangle area. See Methods).

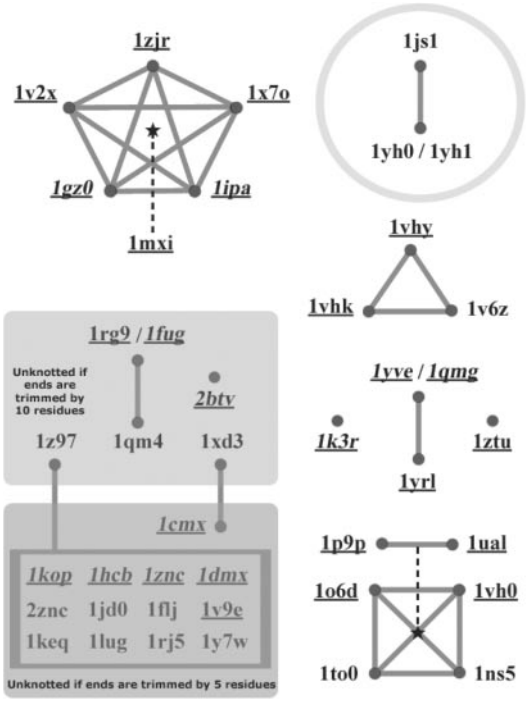


Fig. 4. Relationships between knotted proteins detected by Knotfind. Protein chains are referenced by their PDB codes. Protein pairs sharing sequence similarity (BLASTp evalue < 1E-05) are indicated by solid lines. Structural similarity (MAMMOTH evalue < 1E-07) is indicated by dotted black lines. The pair of sequence similar proteins circled in the top right corner represents a knotted protein fold that has not been previously reported. The 12 chains in the box on the lower left are all sequentially similar to each other. Knotted chains that become unknotted when both ends are trimmed by five residues are grouped in the shaded lower left corner. The shaded box above it contains chains that become unknotted when the ends are trimmed by ten residues. Underlined PDB codes have been previously reported as knotted in the articles describing the experimental structure determination (Badger *et al*., 2005; Lim *et al*., 2003; Elkins *et al*., 2003; Komoto *et al*., 2004; Ahn *et al*., 2003; Nureki *et al*., 2004; Saito *et al*., 2004; Mosbacher *et al*., 2005; Tyagi *et al*., 2005; Pleshe *et al*., 2005; Wagner *et al*., 2005). Articles describing experimental structure determination have not yet been published for 1lug, 1ns5, 1to0 or 1v6z. PDB codes in italics indicate proteins reported by Taylor as being knotted (1cmxA, 1dmxA, 1fugA, 1hcb, 1kopA, 1yveI, 1zncA, 2btvB in Taylor, 2000) (1ipaA, 1k3r, 1qmgA in Taylor *et al*., 2003a) and 1gz0 (Taylor *et al*., 2003b). Note that (Taylor *et al*., 2003a) additionally reports six ''accession numbers for knotted proteins'' that were not found to be knotted here either by the Knotfind algorithm or by visual inspection. One case, 1g0z, is likely a typographical error for 1gz0, which is later reported as knotted in (Taylor *et al*., 2003b). The other five proteins, 1mt6, 1mvh, 1h3i, 1ml9, and 1mlv were later reported to not contain true knots according to the algorithm of Taylor in (Taylor *et al*., 2003b).

the Knotfind algorithm. To minimize the possibility of such false positives, triples are considered for simplification in order of the *i*-1, *i*+1 distance, allowing the simplification to start with the most local backbone features before simplifying more global features. Using this strategy, only one false positive is observed among the 9,557 protein chains examined here. The trapped chain configuration for this chain, 1ogdA, is shown in Figure 2. As described in the Methods above, chains that cannot be fully simplified in the first pass of the algorithm are subjected to a second check during which triples are ordered according the area of the triangle that each defines, considering smallest area first. When used in isolation, this area-based ranking method resulted in four false positives (1e2kA, 1y6vA, 2a65A, 2c5aA). When the two methods are applied sequentially, no false positives are observed in the set of PDB chains examined, or in any of the knotted decoy structures that were visually inspected.

One of the main advantages of the Knotfind algorithm is its speed. Despite using two different triplet-ordering schemes in cases where the first scheme does not result in a completely simplified chain, the algorithm as implemented in Rosetta requires on average less than 0.01 seconds for a single chain. When using Rosetta to evaluate the 9,553 chains from the PISCES server, incorporating the Knotfind algorithm added fewer than 90 seconds to the overall run time on an Intel(R) Xeon(TM) CPU 2.80GH compared to evaluating the chains using Rosetta without Knotfind. Most of the time in Knotfind is spent determining if triples can be simplified by establishing if any line segments intersect the triangles defined by each triple. The number of triples checked depends linearly on the length of the chain in the absence of a knot, while knotted chains require that more triangles be tested for simplification than would be expected on the basis of chain length (Figure 3).

3.2 Knots in protein structures

The Knotfind algorithm was initially applied to protein structures in the RCSB PDB. Twenty-one deeply knotted proteins were found in the collection of 9,553 protein chains taken from the PDB (∼0.2%). In addition, eighteen proteins were identified to contain shallow knots which disappear after trimming five to ten residues from the termini. Of the twenty-one deeply knotted proteins detected,

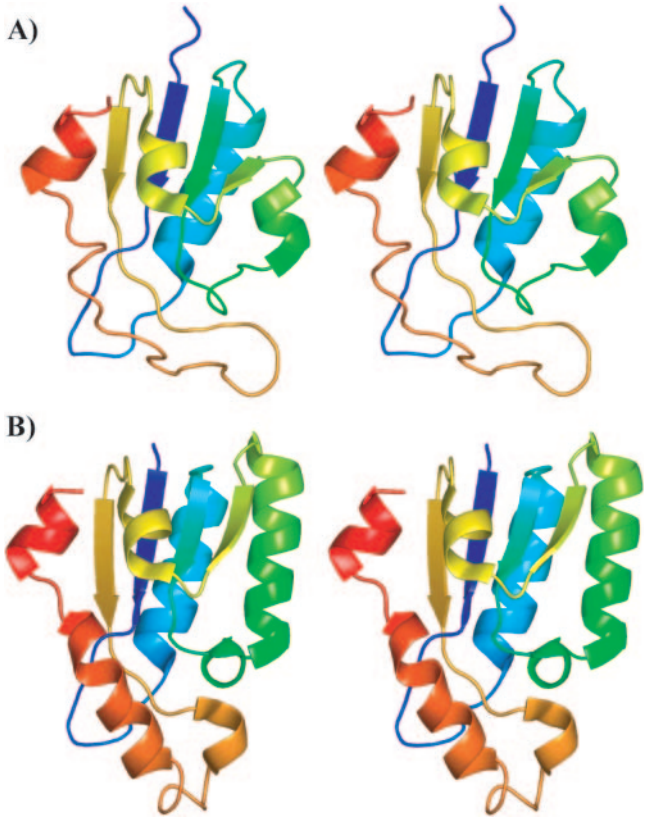


Fig. 5. Stereo view of a previously unreported deep trefoil knot in acetylornithine transcarbamylase. (A) Residues 165-266 of 1js1 chain X (324 residues total) contain a deep trefoil knot where the loop between the yellow strand and red helix threads through the loop comprised of the blue strand and cyan helix. (B) Residues 171-285 of 1yh1 chain A (336 residues total) also contain a deep trefoil knot where the loop between the yellow strand and orange helix threads through the blue loop.

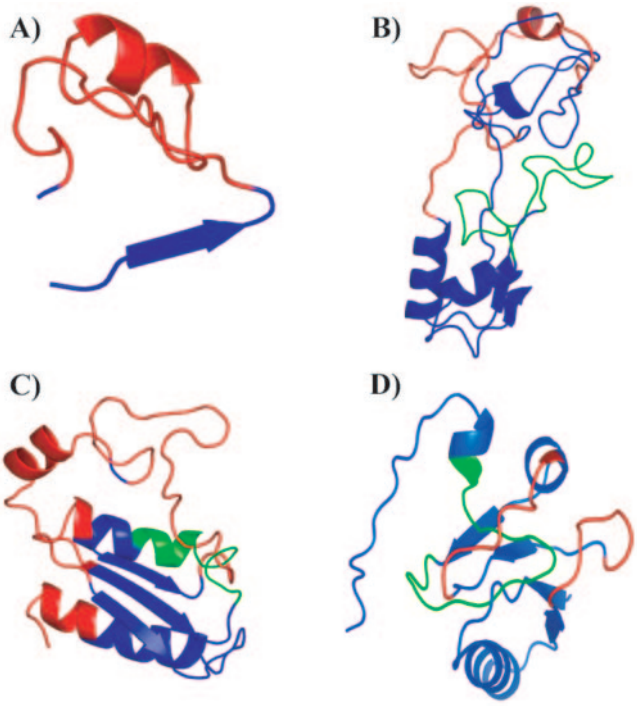


Fig. 6. Examples of knots observed in Rosetta decoys. (A) A Type 1 SVR knot from a T195 CASP 5 decoy (only residues 181-217 are shown), where a knot is entirely localized within a single SVR (residues 188-215, red). The local template structure is shown in blue. (B) A Type 2 SVR knot from T261 (only residues 58-207 are shown), where an SVR (residues 164-189, shown in green) threads through a template region (blue). This model was submitted as Robetta's top ranked model for the target. (C) A Type 3 SVR knot from a T195 CASP 5 decoy (only residues 181-299 are shown), where two SVRs thread through one another. The SVR comprising residues 188-215 in shown in green while the SVR spanning residues 242-253 is shown in red. Template regions are shown in blue. (D) T202 model 1 submitted by Robetta for CASP 6 (only residues 1-101 are shown). An SVR (residues 69-85, shown in green) threads through both the template (blue) and through another SVR (residues 49-56 in red), making this both a Type 2 and Type 3 SVR knot.

most have been previously reported, or a knot in a protein with sequence or structural similarity has been previously reported. Novel deep trefoil knots were detected, however, in two acetylornithine transcarbamylases, 1js1, and 1yh0/1yh1, which are similar in sequence and structure to each other. These proteins are not sequentially or structurally similar to any other previously reported knotted proteins (Figure 4). The trefoil knot in 1js1 and 1yh1 is shown in stereo in Figure 5.

Interestingly, the knot in aceytlornithine transcarbamylase is found in the acetylornithine binding domain, where two loops, a proline rich loop and the 240s loop, appear to be threaded through one another (residues 173-183 and 236-259 respectively in 1js1 (Shi *et al*., 2002); residues 177-188 and 252-278 respectively in 1yh0/1yh1 (Shi *et al*., 2005)). These two loops are presumably responsible for specificity for acetylornithine relative to the unacetylated substrates preferred by the structurally similar, but unknotted, enzymes ornithine transcarbamylase (36% sequence similarity) and aspartate transcarbamylase (40% sequence similarity). The 240s loop in acetylornithine transcarbamylase lacks the essential binding motifs found in the ornithine and aspartate transcarbamylases. Shi *et al*., (2002) hypothesize that the conformational rigidity of the proline-rich loop, which contains four prolines not found in ornithine transcarbamylase, may be responsible for excluding ornithine from the active site by preventing movement of the 240s loop towards the active site.

3.3 Knots in Rosetta decoys

Approximately 5% of the CASP 5 decoys were found to have knots (2,163/45,366). During the course of CASP 5, a high frequency of occurrence of knots had been observed for certain targets, requiring a significant effort in manual inspection to discard those models containing knots (Rohl *et al*., 2004a). This non-uniform distribution of knotted decoys was confirmed, as some targets showed a high percentage of knotted conformations, while others had virtually none (Figure 7A).

To gain a better understanding of the origin of knots in CASP 5 decoys, we also manually inspected the 291 knotted decoys for target T195 which showed the highest frequency of knot formation. SVRs judged to be responsible for knot formation fell into one of three different categories (Figure 6): 1) a single SVR contained a knot that was entirely localized to this SVR (4 examples) 2) a SVR

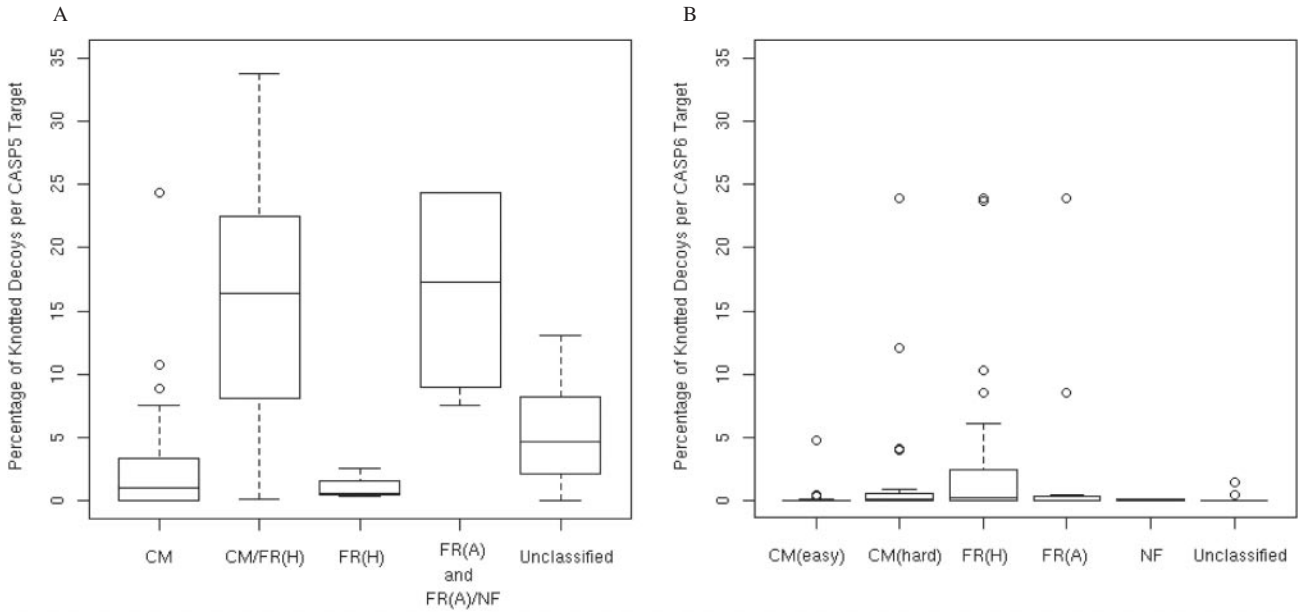


Fig. 7. Frequency of knotted decoys using Rosetta for targets in CASP. The frequencies of knots in decoys sets for A) CASP 5 and B) CASP 6 targets are shown as boxplots. Targets have been binned by difficulty using the assignments defined by CASP assessors (Kinch *et al*., 2003; Tress *et al*., 2005b). In cases where multiple domains of one CASP target have different classifications, the decoy set for the target is included in each classification. Categories, in order of generally increasing difficulty, are comparative modeling (CM); fold recognition, homologous (FR(H)); fold recognition, analogous (FR(A)); and new fold (NF).

threaded through a template region (157 examples), and 3) two SVRs wrapped around one another (138 pairs). In this analysis, SVRs of three residues or less were treated as part of the fixed template due to the fact that their conformation is highly constrained by the geometric constraints imposed by the template.

Additionally, we analyzed CASP 6 models submitted by the automated Robetta server in which the methods used in CASP 5 Rosetta CM predictions were implemented (Bradley *et al*., 2003). Robetta submitted three models in CASP 6 that contained knots which it ranked as its best predictions for T202, T203, and T261 (Figure 6). Seven additional Robetta models which were ranked below the top model also contained knots (T199 Model 3; T202 Model 5; T211 Model 2; T208 Model 2; T235, Domain 1, Model 2 and 4; T261 Model 2). Visual analysis of these structures was consistent with the hypothesis that knot formation was related to SVR modeling, and all knots in Robetta models could be classified as Type 2 and Type 3 as defined above.

Based on our analysis of CASP 5 decoys, we modified our SVR modeling procedure in an attempt to reduce the frequency of knot formation in CASP 6 decoy sets. Libraries of conformations were generated for each SVR and these libraries were screened to eliminate any conformations that resulted in knots when grafted onto the template structure in the absence of all other SVRs. Applying this procedure to T195, we found that pre-filtering the conformational libraries reduced the frequency of knots from 25% (971/3,966) to approximately 20% (745/3,737) by detecting all Type 1 and Type 2 SVRs. Additionally we applied this modified protocol in CASP 6 predictions and found decrease in the overall frequency of knotted decoys (∼1%, 1,343/119,543) relative to that observed for CASP 5 decoys (5%). Knot frequencies in individual decoy sets are shown in Figure 7.

3.4 Knots in Undertaker CASP 6 decoys

In order to assess the extent to which knot formation is specific to the modeling strategy used by Rosetta, we also examined decoy sets generated by the SAM-TO4 group (Group 166) method for CASP 6 targets using the Undertaker program (Karplus *et al*., 2005). In CASP 6, most decoys created by the Undertaker program were knot-free, but decoys sets for a few targets had a high frequency of knots. The highest occurrence of knots in Undertaker decoy sets was found for T228 (12% of decoys knotted), T237 (9% knotted) and T218 (8%). On these same targets, the Rosetta decoy sets had knot frequencies of 10%, 6%, and 0%, respectively. In general, however, there was little or no correlation between the knot frequency in Undertaker decoys and Rosetta decoys across the CASP 6 targets (unpublished data). The Undertaker method does not explicitly model regions of the backbone as either part of a template or a SVR. However, it is similar to Rosetta in that it introduces chain breaks at points corresponding to gaps in the alignment. Visual inspection of Undertaker decoys indicated that the majority of the knots in Undertaker decoys could be explained by threading that occurred while resolving gaps in the backbone or when merging two domains that were modeled separately (K.Karplus, personal communication).

4 DISCUSSION

4.1 Efficacy of the knotfind algorithm

An algorithm for knot detection has been previously described by Taylor (2000) and applied to detect knots in protein structures in the PDB. Taylor's algorithm progressively smoothes the protein backbone: at each iteration, each atom in the backbone is moved

incrementally toward the midpoint of the line segment formed by the N- and C-terminally adjacent atoms, subject to a clash check that ensures the protein backbone does not pass through itself. Knotfind shares the basic approach of trying to straighten out the protein chain, but does so in a stepwise fashion that avoids the need to compute new atom positions and enables the algorithm to converge rapidly.

An additional benefit of not modifying atomic coordinates during the course of the algorithm is that when a knot is detected in a protein chain, the knot can be localized in the structure without the need to interpret a smoothed or distorted chain trace. In cases where chains cannot be completely simplified to an extended segment, the coordinates of the remaining Cα atoms can be used to facilitate the visual identification and analysis of the knot.

One caveat with the Knotfind algorithm is that its performance with respect to false positives and false negatives has not been rigorously proven. The triplet-ordering schemes used here are selected to attempt to minimize the possibility of false positives by first simplifying local backbone features. Notably, triplet-ordering schemes that do not target local backbone features preferentially over global features tend to result in higher occurrence of false positives. For example, considering triples from N- to C-terminal order results in seven false positives (1e2kA, 1e2wA, 1k7hA, 1ohfA, 1p6xA, 1y6vA, 2a65A). Combining two triplet ordering schemes eliminates all false positives in the set of PDB chains examined here, suggesting that false positives, while possible, are likely to be rare.

4.2 Application of knotfind to Rosetta homology-based structure prediction

The detailed analysis of knotted Target 195 decoys suggests that three sources of knots can be generated by Rosetta's comparative modeling approach: SVRs that knot with themselves, SVRs that thread themselves with the template, and pairs of SVRs that thread through each other. The first type of knot is likely introduced by the high gap penalty used to ensure chain continuity. In our experience, the introduction of such knots is rare, perhaps not surprisingly as significant steric clashes generally accompany such knots. Reductions in the gap penalty accompanied by more efficient methods of loop closure, such as the cyclic coordinate descent method (Canutescu and Dunbrack, 2003) can be used to reduce the likelihood of introducing such knots during the modeling process.

Knots of Type 2 and Type 3 are not localized to a single region of the backbone, but instead are attributed to one section of the chain threading through another. In the Rosetta-based method, such knots can be introduced into models because SVR conformations are selected from databases or are initially modeled only in the context of local stem geometry. When such conformations are combined with a fixed template structure, or with models for other SVR regions, threadings can occur which are difficult or impossible to resolve. To reduce the occurrence of such knots, we filtered libraries of SVR conformations during the generation of CASP 6 targets in order to eliminate conformations that were threaded through the fixed template structure and observed a significant reduction in knotted percentage. Interestingly, in some cases such filtering could also be used to guide alignment choice. For example, if all or nearly all conformations for a particular SVR, selected on the basis of fitting the geometric restraints imposed by the template, result in a knot, the original alignment to the parent structure is likely incorrect as it implies structurally unfeasible gaps.

While the frequency of knots was significantly reduced by filtering with the Knotfind algorithm in CASP 5 compared to CASP 6, decoy sets for some CASP 6 targets still show significant occurrence of knots. Since this filtering step only considered single SVRs in the context of the fixed template, knots that are introduced by pairs of SVRs threading through one another (Type 3), are not detected and are expected to still occur in CASP 6 decoy sets. For CASP 6 predictions, these knotted decoys were eliminated from the final decoy population in the model selection process using the Knotfind algorithm. Such knots however, could be eliminated earlier in the modeling process by pairwise examination of SVR conformations in the libraries, or by checking complete models early in the optimization process.

4.3 General application to structure prediction

The Knotfind algorithm can be applied to the benefit of many structure prediction approaches. The most obvious application of the Knotfind algorithm is the screening of final models to ensure that a knotted decoy is not selected. Such screening is particularly important in an automated method such as Robetta where an expert does generally not examine final predictions manually. The speed of the Knotfind algorithm makes it appropriate not just for post-filtering decoy populations to eliminate knotted structures, but also for application during the protein structure prediction process, either as a filter as described here or as part of a scoring scheme used during optimization. For example, Knotfind is now implemented in Undertaker as a cost function that is only used when the potential for knots is high as determined by an expert predictor.

The causes of high knot frequency in some modeling problems is likely to be specific to the particular method used and the structural details of the protein being modeled. On the basis of comparison of knot formation in Rosetta and Undertaker decoys, it seems likely that the introduction of chain breaks during the modeling process is a contributing factor to increased probability of knot formation. Additionally, the location of such chain breaks and the size of the gap introduced at each discontinuity are likely to be important factors as well. This conclusion suggests that a knot detection algorithm is likely not only to be applicable to homology based methods that must model gaps implied by alignments, but in any protein modeling method that introduces chain breaks during the modeling process, including for example *de novo* prediction methods that have recently been demonstrated to be capable of prediction accuracies better than 1Å for small proteins (Bradley *et al*., 2005).

ACKNOWLEDGEMENTS

We thank Josue Samayoa, David Bernick, and Craig Lowe for the decoy structures predicted for CASP 6, Dylan Chivian and David Baker for CASP 5 decoy sets, and Kevin Karplus for Undertaker decoy sets and discussion.

REFERENCES

Ahn,H.J. *et al*. (2003) Crystal structure of tRNA(m(1)G37)methyltransferase: insights into tRNA recognition. *EMBO J*., **11**, 2593–603.

Badger,J. *et al*. (2005) Structural analysis of a set of proteins resulting from a bacterial genomics project. *Proteins*, **60**, 787–796.

Berman,H.M. *et al.* (2000) The Protein Data Bank. *Nucleic Acids Research*, **28**, 235–242.

Bradley,P. *et al.* (2003) Rosetta predictions in CASP 5: successes, failures, and prospects for complete automation. *Proteins*, **53** (Suppl 6), 457–468.

Bradley,P. *et al.* (2005) Toward high-resolution de novo structure prediction for small proteins. *Science*, **5742**, 1868–71.

Canutescu,A.A. and Dunbrack,R.L. (2003) Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Sci.*, **5**, 963–72.

Chivian,D. *et al.* (2003) Automated prediction of CASP-5 structures using the Robetta server. *Proteins*, **53**, 524–533.

Chivian,D. *et al.* (2005) Prediction of CASP6 structures using automated Robetta protocols. *Proteins*, **61** (Suppl 7), 183–92.

Elkins,P.A. *et al.* (2003) Insights into Catalysis by a Knotted TrmD tRNA Methyltransferase. *J. Mol. Biol.*, **333**, 931–949.

Karplus,K. *et al.* (2005) SAM-T04: what is new in protein-structure prediction for CASP6. *Proteins*, **61** (Suppl 7), 135–142.

Kim,D.E. *et al.* (2004) Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res.*, **55**, 656–677.

Kinch,L.N. *et al.* (2003) CASP5 target classification. *Proteins*, **53** (Suppl 6), 340–51.

Komoto,J. *et al.* (2004) Crystal structure of the S-adenosylmethionine synthetase ternary complex: a novel catalytic mechanism of s-adenosylmethionine synthesis from ATP and MET. *Biochemistry*, **43**, 1821–1831.

Lim,K. *et al.* (2003) Structure of the YibK methyltransferase from Haemophilus influenzae (HI0766): a Cofactor Bound at a Site Formed by a Knot. *Proteins*, **51**, 56–67.

Mansfield,M.L. (1994) Are there knots in proteins. *Nat Struct Biol.*, **1**, 213–214.

Mosbacher,T.G. *et al.* (2005) Structure and function of the antibiotic resistance-mediating methyltransferase AviRb from Streptomyces viridochromogenes. *J Mol Biol.*, **3**, 535–45.

Moult,J. *et al.* (1995) A large-scale experiment to assess protein structure prediction methods. *Proteins*, **23**, ii–v.

Nureki,O. *et al.* (2002) An enzyme with a deep trefoil knot for the active-site architecture. *Acta Crystallogr D Biol Crystallogr.*, **58**, 1129–1137.

Nureki,O. *et al.* (2004) Deep Knot Structure for Construction of Active Site and Cofactor Binding Site of tRNA Modification Enzyme. *Structure*, **4**, 593–602.

Pleshe,E. *et al.* (2005) Structure of a class II TrmH tRNA-modifying enzyme from Aquifex aeolicus. *Acta Crystallograph Sect F Struct Biol Cryst Commun.*, **61**, 722–728.

Richardson,J.S. (1997) Beta-Sheet topology and the relatedness of proteins. *Nature*, **268**, 495–500.

Rohl,C.A. *et al.* (2004a) Modeling structurally variable regions in homologous proteins with Rosetta. *Proteins*, **55**, 656–677.

Rohl,C.A. *et al.* (2004b) Protein structure prediction using Rosetta. *Methods Enzymol.*, **383**, 66–93.

Saito,R. *et al.* (2004) Structure of bovine carbonic anhydrase II at 1.95 A resolution. *Acta Crystallogr D Biol Crystallogr.*, **60**, 792–5.

Shi,D. *et al.* (2002) Crystal structure of a transcarbamylase-like protein from the anaerobic bacterium Bacteroides fragilis at 2.0 A resolution. *JMB*, **320**, 899–908.

Shi,D. *et al.* (2005) Crystal Structure of *N*-Acetylornithine Transcarbamylase from *Xanthomonas campestris*. *J. Biol. Chem.*, **280**, 14366–14369.

Taylor,W.R. *et al.* (2000) A deeply knotted protein and how it might fold. *Nature*, **406**, 916–919.

Taylor,W.R. *et al.* (2003a) Protein knots: A tangled problem. *Nature*, **421**, 25.

Taylor,W.R. *et al.* (2003b) A knot or not a knot? SETting the record 'straight' on proteins. *Comput Biol Chem.*, **27**, 11–15.

Tramontano,A. *et al.* (2001) Analysis and assessment of comparative modeling predictions in CASP4. *Proteins*, **45** (Suppl 5), 22-38.

Tress,M. *et al.* (2005a) Assessment of predictions submitted for the CASP6 comparative modeling category. *Proteins*, **61** (Suppl 7), 27–45.

Tress,M. *et al.* (2005b) Domain definition and target classification for CASP6. *Proteins*, **61** (Suppl 7), 8–18.

Wang,G. *et al.* (2003) Jr. PISCES: a protein sequence culling server. *Bioinformatics*, **19**, 1589–1591.

Tyagi,R. *et al.* (2005) The crystal structure of a bacterial class II ketol-acid reductoisomerase: domain conservation and evolution. *Protein Sci.*, **14**, 3089–3100.

Wagner,J.R. *et al.* (2005) A light-sensing knot revealed by the structure of the chromophore-binding domain of phytochrome. *Nature*, **438**, 325–331.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e260–e270
doi:10.1093/bioinformatics/btl221

Annotating proteins by mining protein interaction networks

Mustafa Kirac[1,*], Gultekin Ozsoyoglu[1] and Jiong Yang[1]

[1]Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH, U.S.A.

ABSTRACT

Motivation: In general, most accurate gene/protein annotations are provided by curators. Despite having lesser evidence strengths, it is inevitable to use computational methods for fast and a priori discovery of protein function annotations. This paper considers the problem of assigning Gene Ontology (GO) annotations to partially annotated or newly discovered proteins.

Results: We present a data mining technique that computes the probabilistic relationships between GO annotations of proteins on protein-protein interaction data, and assigns highly correlated GO terms of annotated proteins to non-annotated proteins in the target set. In comparison with other techniques, probabilistic suffix tree and correlation mining techniques produce the highest prediction accuracy of 81% precision with the recall at 45%.

Availability: Code is available upon request. Results and used materials are available online at http://kirac.case.edu/PROTAN

Contact: kirac@case.edu

1 INTRODUCTION

In this paper, we consider the problem of assigning Gene Ontology (GO) (Gene Ontology Consortium, 2004) annotations to newly discovered proteins. The GO Consortium has produced a controlled vocabulary for protein function annotation that is used in numerous organism-specific protein databases (GO, http://www.geneontology.org). However, presently not all known proteins are annotated in these databases, while many others are only partially annotated.

In general, the most accurate gene/protein annotations are provided by curators who search the literature for articles containing evidence for a particular annotation. Despite having lesser evidence strengths, it is inevitable to use computational methods such as text mining, statistical gene expression analysis and sequence similarity, for fast and a priori discovery of protein function annotations. Currently, the primary method for GO function assignment to proteins is sequence similarity analysis which needs homologs in biological databases (Deng *et al*., 2004), and transferring functional assignments between proteins with low sequence identity (below 40%) is found to be unreliable (Letovsky *et al*., 2003). Recently several successful text mining-based annotation prediction tools (Izumitani *et al*., 2004; Asako *et al*., 2005) have been developed. This approach however needs text parsing and metadata extraction from publications in the literature that describe the functionality of a target protein, a difficult task on its own. As an alternative to the text mining approach, recent work (Troyanskaya *et al*., 2003; Samanta and Liang, 2003; Deng *et al*., 2004; Vazquez *et al*., 2003) has shown that employing a combination of GO annotation and protein-protein interaction (PPI) data is also reasonably effective for accurate prediction of GO annotations for non-annotated proteins.

In this paper, we present a data mining technique that, using protein-protein interaction data, identifies probabilistic relationships between GO annotations of proteins and annotates target proteins with highly correlated GO terms of other proteins. The motivation for our approach comes primarily from the recent discovery (Poyatos and Hurst, 2004; von Mering *et al*., 2003) that the relationship between proteins in a protein interaction network is not only limited to protein pairs (i.e., interaction edges), but also generalizes to functional modules that are not necessarily protein complexes. It is now believed (Hu *et al*., 2005; Sharan *et al*., 2005) that proteins in the same functional module have the same (or similar) functional annotation. Earlier work (Troyanskaya *et al*., 2003; Samanta and Liang, 2003; Deng *et al*., 2004; Schwikowski *et al*., 2000; Hishigaki *et al*., 2001; Vazquez *et al*., 2003) formalized the protein function prediction problem differently: they all considered known protein functions (e.g., GO annotation) as predefined protein classes, and then employed topological features of protein interaction networks to classify proteins and to assign the same function to all proteins in the same class.

Our approach in this paper is to compute the probabilistic significance of GO annotation sequences obtained from the annotations of a sequence of proteins in a protein-protein interaction network. We develop and evaluate two significance analysis techniques: (a) correlation mining for annotation pairs (i.e., GO annotation sequences of length 2), (b) variable-length Markov model for annotation sequences of arbitrary length. After identifying significant annotation sequences, we predict the annotation of a protein as follows. (i) Generate (via random walk) GO annotation sequences where the non-annotated protein (i.e., target protein which is partially or not annotated) interacts with the protein at the tail of the corresponding protein sequence. (ii) Expand each GO annotation sequence by adding a GO term to the end of the GO annotation sequence. (iii) Pick the suffix GO term of the most significant candidate GO annotation sequence as the GO term prediction for the non-annotated protein. Our cross-validation prediction experiments with pre-annotated proteins recovered correct annotations of proteins with 81% precision with the recall at 45%.

Experimentally, we have evaluated the effects of (a) dataset selection, (b) GO sub-ontology selection, (c) defining random walk sampling size and (d) setting maximum GO annotation

*To whom correspondence should be addressed.

sequence length on the accuracy of our predictions. In our experiments, highest prediction accuracy is obtained with correlation mining on BIND dataset (BIND, http://www.bind.ca) (vs. other datasets using GO as function annotations). Among the three sub-ontologies of GO (i.e., biological process, cellular component and molecular function), cellular component ontology produced the highest prediction accuracy. To compare our results with previous work (Deng *et al*., 2002; Schwikowski *et al*., 2000; Hishigaki *et al*., 2001), our prediction methodology performed better than the results of known methods Markov random fields (Deng *et al*., 2002), neighbor-counting (Schwikowski *et al*., 2000) and chi-square (Hishigaki *et al*., 2001) by 6.6%, 31% and 19.7% respectively.

Our work differs from the previous work in two aspects. First, the previous research on protein function prediction focuses on a particular protein function set, and builds models based on the direct interactions of proteins (Troyanskaya *et al*., 2003; Samanta and Liang, 2003; Deng *et al*., 2004; Schwikowski *et al*., 2000; Hishigaki *et al*., 2001; Vazquez *et al*., 2003). In comparison, we mine the complete protein interaction network to locate relationships between protein functions (i.e., in our case, GO terms). In other words, we assign a GO term annotation to a protein P if the annotation is implied by the existing GO term annotation patterns (i.e., annotation sequences) of proteins that interact with P. Since the source of protein interaction data mostly comes from unverified high-throughput experiments, protein interaction data contains many false positives (Deng *et al*., 2003). Our prediction of a GO term (function) requires a statistically significant usage of that GO term in a particular pattern. Therefore our methods are not affected by false interactions/false annotations as long as the corrupt data does not span a major portion of the interaction data.

Other works that apply patterns (a.k.a., motifs) to infer functions in protein interaction networks view those patterns as clusters, and distribute the most significant function in a cluster to non-annotated proteins (Hu *et al*., 2005; Sharan *et al*., 2005). This method successfully predicts the annotation of proteins that build a protein complex since all the proteins in the complex have the same function. However, it does not offer any prediction for the annotation of a protein which is not part of a frequent protein interaction motif. In contrast with (Hu *et al*., 2005; Sharan *et al*., 2005), our approach can predict the function of a protein that interacts with at least one annotated protein by using annotations of the proteins as well as the topological features of protein interaction networks.

The rest of the paper is organized as follows. In Section 2, we give a brief overview of our methodology. In Section 3 we describe our GO function prediction algorithms. In Section 4, we experimentally evaluate our GO function prediction algorithms. Section 5 lists the related work. Finally, in Section 6 we give a summary of our results.

2 METHODS

In protein interaction networks, Hishigaki *et al*. (2001) and Schwikowski *et al*. (2000) note that if interaction partners of a protein P are annotated with a certain functionality then, with some probability, P is also annotated with the same functionality. This probability can be used to infer GO functions of non-annotated proteins. Others (King *et al*., 2003) found correlations between GO annotations of proteins, and developed probabilistic techniques to extend known annotations of proteins with additional GO terms. The same approach with (King *et al*., 2003) can be applied to annotations of proteins spanning over several proteins in a protein interaction network. We integrate, in this paper, (i) the probabilistic significance of *GO annotation sequences* (i.e., a sequence of GO terms that corresponds to the annotations of a sequence of proteins in a protein-protein interaction network) on protein interactions and (ii) correlation of GO terms in protein annotations into a GO term prediction model.

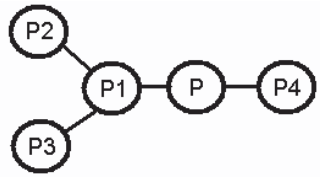


Fig. 1. Protein interaction network example.

We generalize the relationships between occurrences of GO terms in a protein interaction network. We make the same assumption of (Schwikowski *et al*., 2000; Hishigaki *et al*., 2001) that the probability of assigning a GO term to a protein depends on the GO term annotation of neighbor proteins. Moreover, to differentiate between the near and far neighbors, we model neighborhood information of a protein in the form of annotation sequences where prefixes of annotation sequences represent far neighbors, and suffixes of annotation sequences represent near neighbors.

Let $\pi_{i,t}$ = Prob ($t \in goann(P_i) \mid T \in goann(N\text{-}P_i)$) be the probability that protein P_i is annotated with GO term t given the GO term annotations T of all proteins (except P_i) in network N, where *goann*(P) represents the GO term annotation of protein P. Since the annotation of P_i only depends on the annotation of its neighborhood (i.e., proteins having a path to P_i by following a sequence of interactions) rather than the whole protein interaction network, we can compute the same probability as:

$\pi_{i,t}$ = Prob ($t \in goann(P_i) \mid observe(O_1,P_i) \wedge observe(O_2,P_i) \wedge \ldots \wedge observe(O_{k+n+m}, P_i)$). *observe*($O_j$, P_i) represents the event of observing the annotation sequence O_j on protein paths such that the tail protein of O_j interacts with P_i. Observing an annotation sequence on a protein path is described as follows. Let $O_i = a_1,a_2\ldots a_n$ be an annotation sequence where a_j (for $1<j<n$) is a GO annotation of protein P_j in the protein path $\rho = P_1, P_2\ldots P_n$. O_i is an annotation sequence observation of P_i, if P_i interacts with P_n. We give an example.

Example 1: In Figure 1, protein P has 3 distinct protein paths, namely, P2-P1, P3-P1 and P4. Let O_i be an annotation sequence observation at protein P, and $O_1\ldots O_k$ be the annotation sequences corresponding to the protein path P2-P1, and $O_{k+1}\ldots O_{k+n}$ and $O_{k+n+1}\ldots O_{k+n+m}$ be annotation sequences corresponding to protein paths P3-P1and P4, respectively. Then, the probability of P having the GO term annotation t becomes:

$$\text{Prob } (t \in goann(P) \mid observe(O_1, P_i) \wedge observe(O_2, P_i) \wedge \ldots \wedge observe(O_{k+n+m}, P_i))$$

Individual observation probabilities, Prob (*observe*(O_1,P_i)), Prob (*observe*(O_2, P_i)),..., Prob (*observe*(O_1,P_i)) are not independent since they are all observed on the same protein. As a result, there is no easy way to compute $\pi_{i,t}$. We approximate $\pi_{i,t}$ as an aggregation:

$$\pi_{i,t} \approx \theta \begin{pmatrix} \text{Prob}(t \in goann(P_i) \mid observe(O_1)), \\ \text{Prob}(t \in goann(P_i) \mid observe(O_2)), \\ \ldots, \\ \text{Prob}(t \in goann(P_i) \mid observe(O_n)) \end{pmatrix},$$

where θ is an aggregation function. The conditional probability Prob($t \in goann(P_i) \mid observe(O_j,P_i)$) can be approximated as $v(O_j t)/v(O_j)$, where v(S) is the number of unique protein paths in protein interaction network N that is annotated with the GO annotation sequence S (i.e., the frequency of the annotation sequence S in the protein interaction network), as all proteins are equally likely to have the same GO term annotation as long as they exhibit the same annotation sequences on their neighborhood, according to the assumption that the probability of assigning a GO term to a protein depends on the GO term annotations of neighboring proteins.

To compute the probability $\pi_{i,t}$, we first count the frequencies of possible annotation sequences. Computing real frequencies of annotation sequences is computationally infeasible due to the exponential number of protein paths and annotation sequences. Thus, we reduce the number of GO terms by eliminating the ''uninformative'' GO terms (i.e., GO terms assigned to a small number of proteins). Next, we approximate the frequencies of annotation paths by sampling a sufficient number of annotation sequences. In our experiments, we found that increasing the sample size does not significantly increase the accuracy of prediction if the sample size is sufficiently large (see Section 4.4). We store the frequencies of annotation sequences in a structure called the *probabilistic suffix tree* (PST) (Yang and Wang, 2003). A PST is a trie with node and edge labels, and a counter at each node which represents the frequency of the corresponding annotation sequence. The PST allows us to keep the frequency of variable-length protein paths, and to compute the probability of a GO term, given an annotation sequence. A probability-distribution-comparison-measure (i.e., a ''divergence'' measure) is used in the PST to check whether the following holds:

$$\text{Prob}\,(t \in goann(P_i) \,|\, observe(O_j, P_i)) \approx \text{Prob}\,(t \in goann(P_i) \,|\, observe(O_j^k, P_i))$$

where O_j^k is a suffix of O_j of length k (to determine that increasing k is not worth the effort).

To predict the annotation of a given non-annotated protein P using the PST, we use the following procedure. Using random walk technique, we sample a sufficiently large number of annotation sequences whose tail is the annotation of protein P, and therefore, marked as unknown. Next, we run the known prefixes of the annotation sequence samples on the PST to compute a probability distribution of GO term annotations corresponding to each annotation sequence. Finally we aggregate all probability distributions to obtain an annotation prediction set, and pick top k annotations from the set. See Section 3.2 for details.

For annotation sequences of length 2 (i.e., annotation pairs) we employ correlation mining technique (He *et al*., 2004) since it is feasible to employ all GO terms, rather than a subset of it. We build correlation measures using the frequencies of co-appearing GO terms assigned to a pair of interacting proteins. After computing interaction-based correlation between all possible GO term pairs (see Section 3.1.1 for details), we make a GO annotation prediction for protein P as follows. We generate a set of GO terms by inserting the GO annotation of all interaction partners of P into a set S. For each GO term t_i in S, we obtain correlation values between t_i and all other GO terms, and we form a correlation vector V_i whose each dimension corresponds to the correlation between a GO term and t_i. Each correlation vector V_i represents the effect of GO term t_i on prediction of GO annotations for P, based on the observations made on the training set. Hence, aggregation V of all correlation vectors $V_1, V_2, \ldots, V_n$ reflect the effects of all GO terms in S. Finally we pick as our GO annotation prediction set the top k GO terms with highest correlation values in V (see Section 3.1).

We also apply correlation mining on the GO annotation of proteins without incorporating the protein interaction information. In this case, *two GO terms are highly correlated* if they occur together in several protein GO annotations. We employ the annotation-based correlation of GO terms to improve the prediction scores obtained as a prediction probability (from PST) or as a prediction correlation value (from interaction-based correlation mining). Annotation of protein P by the GO term t_1 may increase the probability of P being annotated by GO term t_2 when GO terms t_1 and t_2 are highly annotation-correlated. Therefore, if GO terms t_1 and t_2 are highly annotation-correlated and t_2 has a lower prediction score than t_1, we increase the prediction score of t_2 (to a value not higher than the prediction score of t_1) with respect to the strength of annotation-based correlation between t_1 and t_2. See Section 4.6 for the details of prediction score improvement using annotation-based correlation values.

In Section 4, we experimentally evaluate the effect of using PST versus correlation mining to see if distant neighbors of a protein P have an effect on P's annotation. We also evaluate the prediction accuracy improvements when annotation-based correlation values are employed.

	t_j	"no t_j"	Σ
t_i	$F_{11} = C_{ij}$	$F_{10} = C_{i+} - C_{ij}$	$F_{1+} = F_{11} + F_{10}$
"no t_i"	$F_{01} = C_{+j} - C_{ij}$	$F_{00} = C_{++} - C_{ij}$	$F_{0+} = F_{01} + F_{00}$
Σ	$F_{+1} = F_{11} + F_{01}$	$F_{+0} = F_{10} + F_{00}$	$F_{++} = F_{1+} + F_{0+}$

Fig. 2. Computing the contingency table from the frequency table for all terms.

3 ALGORITHMS

3.1 Correlation between GO term pairs

Genes/Proteins sharing common function annotations are found to be genetically related (Tong *et al*., 2004). As a result, recent work on protein function prediction (Schwikowski *et al*., 2000; Hishigaki *et al*., 2001; Deng *et al*., 2002; Deng *et al*., 2004) treats each protein function (e.g., GO terms, FunCat classification) independently, and determines the function of a protein depending on the distribution of the function on the neighbors of the protein. Generally, a protein having one function does not prevent it from having other functions. Therefore, the available techniques are unbiased while predicting protein functions. However, for GO annotations, there are correlations between protein function annotations. A protein being annotated by the GO term A may imply an increase in the probability of the protein being annotated by GO term B when GO terms A and B are highly correlated (King *et al*., 2003). Here, we incorporate the correlation information into a generalized model, and use correlation mining (He *et al*., 2004) to assign GO terms to proteins. In this section, we discuss two different correlation types for GO terms, namely (a) interaction-based-correlation which is the correlation between two GO terms that annotate two separate interacting proteins and (b) annotation-based-correlation which is the correlation between two GO terms that annotate the same protein.

3.1.1 Computation of interaction-based GO correlations **Definition (interaction-based co-appearance, co-absence and cross-appearance):** *With respect to a particular protein interaction (P_1, P_2), (a) two GO terms co-appear if one of the GO terms is assigned to P_1 and the other is assigned to P_2, (b) two GO terms are co-absent if none of the two GO-terms are assigned to P_1 or P_2, (c) two GO terms cross-appear if one of the GO terms is assigned to protein P_1 and the other GO term is not assigned to P_2.*

We compute the interaction-based correlation between two GO terms that belong to the same ontology class (e.g., biological process ontology) by using the protein interaction data (e.g., interaction pairs in the BIND dataset) as follows. First, we generate a matrix M_I for each GO sub-ontology (i.e., biological process ontology, molecular function ontology and cellular component ontology) to keep the interaction-based correlation values between GO terms. For simplicity, here we explain the algorithm for a single sub-ontology and a single matrix. Rows and columns of the matrix M_I represent the GO terms of a particular sub-ontology. We fill each cell in matrix M_I with the correlation value between the GO terms corresponding to the cell by using a correlation measure. Theoretically, any correlation measure is a possible candidate for the algorithm (He *et al*., 2004; Tan *et al*., 2002). Basically, we express correlation measure values (see Figure 3 for a list) in contingency tables (He *et al*., 2004) (see Figure 2).

We build a frequency matrix by a single scan on the dataset, and use the frequency matrix to obtain separate contingency tables.

Measure	Formula
H-Measure (He *et al.*, 2004)	$H_P = 1 - (F_{10} * F_{01}) / (F_{+1} * F_{1+})$
Jaccard (Tan *et al.*, 2002)	$J_P = F_{11}/(F_{11}+F_{10}+F_{01})$
Cosine (Tan *et al.*, 2002)	$C_P = F_{11}/ (F_{+1}*F_{1+})^{0.5}$
Support (Tan *et al.*, 2002)	$Sup_P = F_{11}$
Confidence (Tan *et al.*, 2002)	$Conf_P = \max (F_{11}/F_{1+}, F_{11}/F_{+1})$

Fig. 3. A list of correlation measures that are used in the GO term prediction algorithm.

A cell C_{ij} in the frequency matrix denotes the (interaction-based) co-appearance frequency of term pairs. We also have a special row and a special column for the null term to count how many times the terms occur alone. C_{i+} and C_{i+} represent the column and row sums of the frequency matrix, respectively. C_{++} denotes the sum of all cells. Using the frequency table, the contingency table for terms t_i and t_j is computed as shown in Figure 2.

By using the contingency table obtained from the frequency table and a correlation measure (e.g., Jaccard measure; see Figure 3), we compute the interaction correlation value of each GO term pair. F_{11}, F_{01}, F_{10}, F_{00} in the contingency table represent the co-appearance, cross-appearance, cross-appearance and co-absence frequencies of two terms t_i and t_j, respectively. Other frequencies with the plus sign are column and row sums of the contingency table. Next, we place the correlation values for GO term pairs into the correlation matrix M_I. At this stage, a cell in the correlation matrix M_I [i, j] contains the interaction correlation value of two GO terms t_i and t_j.

We discuss performances of different correlation measures (see Figure 3) in Section 4.7.

3.1.2 Computation of annotation-based GO correlations **Definition (annotation based co-appearance, co-absence and cross-appearance):** In terms of GO annotations of a protein P, two GO terms T_1 and T_2 (a) co-appear if both GO terms are assigned to P, (b) are co-absent when none of T_1 and T_2 are assigned to P, (c) cross-appear if only one of T_1 and T_2 is assigned to P.

We compute the annotation-based correlations between GO terms by using GO annotations. This stage is very similar to the computation of interaction-based correlation values. Again, we create matrix M_A where rows and columns of the matrix represent GO terms of a particular ontology. Next, we generate the frequency table by processing all proteins in the dataset. Then we create contingency tables for every pair of GO terms. Finally, we fill each cell in M_A with correlation measure values using the corresponding contingency table.

3.1.3 GO term annotation using correlation mining Our motivation to use interaction-based correlations for GO term annotation: If we obtain highly correlated GO term pairs, we can also predict GO terms of a non-annotated protein *Q*. We know the proteins that interact with *Q*; so we build a set of GO terms as a *base GO term set for Q* by unifying the GO terms of the proteins that interact with *Q*. Using the base GO term set, we generate *a prediction set of Q* by selecting the GO terms that are highly correlated with the base set of *Q*. In Section 4, we empirically evaluate the validity of the claim that the top GO terms in the prediction set correctly annotate the protein *Q*.

We compute GO term prediction scores of a non-annotated protein *P* based only on the values in matrix M_I as follows. Using the protein interaction dataset, we generate a set *S* of proteins that interact with *P*. Then we add the GO terms of each protein in *S* to a GO term set *G*. Note that, repetition of a GO term in *G* is allowed so that the impact of frequent GO terms in the neighborhood is naturally increased. Next, for each term t_i in *G*, we extract the corresponding column from M_I and generate a correlation vector V_i. GO terms to be predicted for *P* must be interaction-correlated with all the terms in *G*. Therefore, each GO term in *G* should contribute to the GO term prediction scores of *P*. So, we sum up all correlation vectors and generate a single vector ϑ as the *GO term prediction score vector* for P. Then we normalize the scores in ϑ (e.g., via dividing the scores by the maximum score) since the number of GO terms in *G* varies by protein to protein. As a result, the final ϑ contains the scores of each GO term determining the prediction quality of each GO term with respect to *P*.

3.2 GO term annotation sequences

In section 3.1, we described a correlation mining technique among GO terms of a protein and its direct interaction partners. In this section we focus on distant neighbors of proteins, build GO term annotation sequences, and compute the likelihood of having a sequence of annotations on a protein interaction path.

The scope of a GO term annotation, namely protein interaction paths, grows exponentially in the size of the interaction network; therefore, our approach is to sample and use only a fraction of all possible protein interaction paths.

In our analysis, we randomly select protein paths and protein annotations to generate a sample of annotation sequences. Our approach is to select protein paths using random walks in which we randomly pick a starting protein, and walk over the graph by randomly selecting the next adjacent protein. We assume that all interactions are equally likely, ignoring the fact that they do not have the same reliability (Letovsky *et al*., 2003). The maximum length of a random walk is not bounded unless explicitly defined (see section 4.4). We prevent loops and infinite-length paths by disallowing repetition of proteins on a path. Each time we finish generating a protein path, we also generate annotation sequences by randomly selecting a single annotation from each protein on the path.

To capture statistical correlations of different lengths, we use a Variable-length Markov Model (VMM) to compute and store likelihoods of the annotation sequences. Hidden Markov Model (HMM) is proven to be a successful tool in the analysis of biological data (Durbin *et al*., 1998). An HMM has a fixed number of states, namely, D states (D-th order Markov model). In our case, we do not know the optimum length of the function annotation sequences. Annotation sequences longer than the optimal length (i.e., using further neighbors of a protein rather than near ones) have less influence on the annotation of a protein that the sequence belongs to. Therefore, one cannot pick a good upper bound D, and design the HMM accordingly. VMMs deal with a class of random processes in which the memory-length varies, in contrast to a D-th order Markov model where the length of the memory is fixed. There are many VMM types and prediction algorithms (Begleiter *et al*., 2004). We select the Probabilistic Suffix Tree as our VMM.

The Probabilistic suffix tree (PST) (Begleiter *et al*., 2004) is a variation of the *suffix tree* (Galil and Ukkonen, 1995) for making predictions using the probabilities assigned to the nodes of PST in the training phase. The traditional *suffix tree* (ST) built for a sequence S is a rooted directed tree where each node represents a suffix of S and each edge represents a symbol concatenated to a

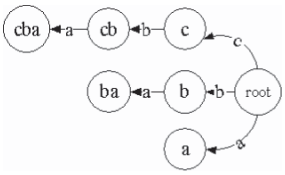


Fig. 4. Suffix Tree for ''cba''.

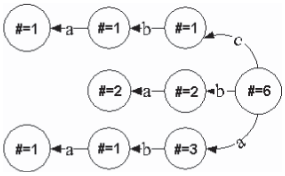


Fig. 5. A GST with counters.

suffix. For each node, concatenating the edge labels from root to a node gives the node label, namely, a distinct suffix of the string S.

The *generalized suffix tree* (GST) is a suffix tree that combines suffixes of a set of strings, $T = \{S_1, S_2, \ldots S_n\}$ (see Figure 4). The PST model further modifies GST, by adding a counter to each node which represents the frequency of the string segment in the string set of GST.

Example 2: Figure 5 shows a PST example built from the training set S = {abc, aba}. We insert all suffixes of reverse strings in the training set to a PST. Therefore we have {cba, ba, a, aba, ba, a} inserted to the tree.

We use the PST to store the frequencies of annotation sequences in a training set obtained via random walks on a protein interaction dataset. We use the frequency information to compute the conditional probability Prob(t | O), i.e., given the annotation sequence O (on a protein path ρ), the probability of having GO annotation t (assigned to the protein P connected to the protein path ρ). Using PST counters, one can compute the conditional probability of a symbol a_n appearing after a given sequence $a_1,a_2,\ldots,a_{n-1}$ as follows:

$$\text{Prob}\ (a_n \mid a_1,a_2,\ldots,a_{n-1}) = v(a_1,a_2,\ldots,a_n)/v(a_1,a_2,\ldots,a_{n-1})$$

where $v(\sigma)$ denotes the frequency of occurrence of segment σ in the training set. Thus, Prob(t | O) is computed as v(O.t)/v(O).

In the PST, we store the shortest *significant* suffixes of training sequences when it is possible to represent the whole sequence with its suffix (see example 3).

Example 3: Let a training set contain 25 occurrences of each sequence ''bc'', ''abc'', ''bd'' and ''abd''. When we use the training sample to compute the probability Prob(c | ab) of having symbol c followed by ab, we compute v(abc)/v(ab) = 25/50 = 1/2 (note that both abd and abc contain ab). When we use the shorter suffix (of length 1), we compute Prob(c | b) and we get v(bc)/v(b) = 50/100 = 1/2 (note that b is contained in all sequences). The probability does not (significantly) change; therefore there is no need to keep extra nodes in the tree for ''abc'' and ''abd'', and keeping ''bc and bd'' are sufficient.

Assume S is a string of symbols defined in the alphabet Σ and the probability of having the symbol x followed by S is Prob $(x \mid S)$. In probabilistic prediction algorithms (Bejerano *et al*., 2001), the aim is to have a close prediction probability Prob′$(x \mid S)$ that is close to Prob $(x \mid S)$. The main idea of VMMs is that if the probability Prob′ $(x \mid yS)$ that predicts the next symbol x followed by yS, is not significantly different than Prob′ $(x \mid S)$, the shorter-length prediction Prob′ $(x \mid S)$ can be also used to estimate Prob $(x \mid S)$. Using only the shortest significant suffix that determines the next symbol reduces the memory and computation requirements of a PST. However, Prob′$(a_n \mid a_1,a_2,\ldots,a_{n\text{-}1})$ cannot always be computed by using the frequency count ratio $v(a_1,a_2,\ldots,a_n)/v(a_1,a_2,\ldots,a_{n\text{-}1})$ since we only store the shortest significant suffixes in PST. Therefore, each conditional probability is computed by using the longest available suffix frequencies in the PST. Here, we obtain

$$\text{Prob}'(a_n \mid a_1,a_2,\ldots,a_{n-1}) = \text{Prob}'(a_n \mid a_k,a_{k+1},\ldots,a_{n-1})\ \textit{and}$$
$$\text{Prob}'(a_n \mid a_k,a_{k+1},\ldots,a_{n-1}) = v(a_k,a_{k+1},\ldots a_n)/v(a_k,a_{k+1},\ldots,a_{n-1}),$$

where $a_k,a_{k+1},\ldots,a_n$ is the longest observed/stored suffix of the sequence $a_1,a_2,\ldots,a_n$ in the PST.

We remove insignificant nodes using the weighted Kullback-Leibler (KL) divergence (Yang and Wang, 2003) to create probability distributions at each PST node. KL divergence is defined as:

$$\Delta H(yS,S) = \text{Prob}'(yS) \sum_x \text{Prob}'(x \mid yS) \log \frac{\text{Prob}'(x \mid yS)}{\text{Prob}'(x \mid S)}$$

where we compare the log ratios of the child node probability distribution (given the longer suffix, Prob′ $(x \mid yS)$) with parent node probability distribution (given the shorter suffix, Prob′$(x \mid S)$). Unless the KL-divergence $\Delta H(yS,S)$ exceeds a predefined threshold σ, we use the shorter suffix S (i.e., the parent node) instead of yS (i.e., the child node), and the node for symbol (i.e., GO term) y at the leaf level is not created or deleted if it already exists.

Example 4: To build a PST for sequences ''abc'' and ''aba''. First we insert ''cba'', ''ba'', ''a'' and ''aba'', ''ba'', ''a'' to empty tree. (See example 2). Then, we compute the probability distributions at each node. For instance, at node 5, we compute the following distribution (See Figure 6):

$$\text{Prob}(a \mid b) = \text{v(ba)/v(b)} = 1/2$$
$$\text{prob(b} \mid \text{b)} = \text{v(bb)/v(b)} = 0/2$$
$$\text{prob(c} \mid \text{b)} = \text{v(bc)/v(b)} = 1/2$$

Next, we smooth the probabilities at the nodes (See Figure 6). For instance at node 5, we have:

$$\text{Prob}(b \mid b) = 0 \rightarrow 0.01$$

Subtract 0.01/2 from the rest of the two probabilities:

$$\text{Prob}(a \mid b) = 1/2 - 1/200 = 99/200$$
$$\text{Prob(c} \mid \text{b)} = 1/2 - 1/200 = 99/200$$

Finally, we remove insignificant nodes from the tree. In Figure 6, the nodes to the left of the boundary line are insignificant nodes (i.e., their probability distributions are not much different from their parents' distributions).

3.2.1 GO Annotation using probabilistic suffix tree After we build the PST using annotation sequences sampled from the training protein interaction network, next we predict the annotation of a non-annotated target protein P as follows. Using the random walk algorithm, we retrieve a protein path sample set Q starting

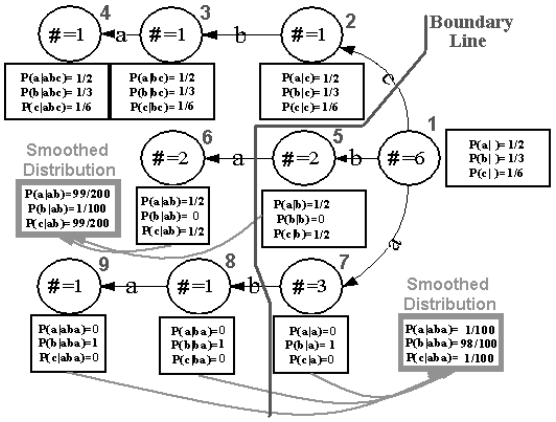


Fig. 6. A PST with probability distributions at nodes (displaying (a) smoothing by redistribution (b) insignificant node elimination by trimming tree with a boundary line).

at the source protein P. Then we remove P from the ends of protein paths in Q, and reverse each protein path in Q. Next, we convert protein path samples Q into annotation sequence samples T by randomly picking a GO function annotation of a protein for each protein path in Q. Then we use the PST to derive the probability distribution of the next symbol for each annotation sequence in T, and form a vector with the values in the probability distribution. Next, we aggregate (i.e., average) all probability distribution vectors to generate a single prediction score vector. Finally, we obtain a list of GO annotation predictions for P by picking only the top GO terms with a prediction score above a given threshold τ.

3.3 Prediction score improvement

In this stage, we employ annotation based correlation values of GO terms to improve the prediction scores (i.e., either PST probability distributions or interaction-based correlation values). Annotation of protein P by the GO term T_1 may increase the probability of P being annotated by GO term T_2 when GO terms T_1 and T_2 are highly annotation-correlated. Therefore, if GO terms T_1 and T_2 are highly annotation-correlated and T_2 has a lower prediction score than T_1, we increase the prediction score of T_2 (to a value not higher than the prediction score of T_1) with respect to the strength of annotation-based correlation between T_1 and T_2.

In our experiments, we computed the prediction accuracy with and without using the prediction score improvement based on annotation-based correlation values. When we enabled score improvement, we obtained up to 30% improvement in our prediction F-values of some proteins (See Section 4.6).

4 EXPERIMENTS AND RESULTS

To build a protein interaction network for our experiments, we have used organism-(i.e., yeast) specific interaction datasets of MIPS (MIPS, http://mips.gsf.de) and GRID (GRID, http://biodata.mshri.on.ca/grid Breitkreutz *et al*., 2003), and complete dataset of BIND. All datasets include both physical and genetic interactions of their scopes. For comparisons of available techniques, we used the dataset of Deng *et al.* (2002) (DENG) and compared our implementations with their prediction results (DENG, http://www-hto.usc.edu/msms/FunctionPrediction). In the DENG dataset,

DATASETS	#proteins	#interactions	#annotations
BIND(BP-MF-CC)	7222	26212	327 (200-88-39)
GRID(BP-MF-CC)	2597	13121	76 (32-21-23)
MIPS	3401	20070	99
DENG(BIO-ROLE-LOC)	1398	3870	129 (57-43-29)

Fig. 7. Dataset details.

proteins are annotated with pre-defined function classes instead of GO terms. The MIPS dataset is annotated with a special function catalog named FunCat (FunCat, http://mips.gsf.de/projects/funcat).

Our experiments with GO term annotation sequences cannot scale to large numbers of GO terms. Therefore, we reduced the number of annotations by picking a subset of the annotations which is referred to as *informative nodes* in (Zhou *et al*., 2002). A GO term is viewed as an informative node in the GO hierarchy: (a) if the number of proteins that are annotated with this node is less than a threshold, namely γ, and (b) if each of the children of the node is annotated with less than γ proteins. We removed from the datasets all GO annotations which are not informative. We picked γ=500 in the BIND dataset and $\gamma = 30$ in the MIPS and GRID datasets. In the DENG dataset, protein function annotations are a flat list of function labels. We directly used DENG data annotations. We also remove from datasets any protein with no annotations or no interaction partners in order to arrange a clean cross validation setting. Final dataset details are listed in Figure 7.

Gene ontology (GO) consists of three graph-structured term vocabularies, namely *biological process ontology* (BP), *molecular function ontology* (MF) and *cellular component ontology* (CC) (Gene Ontology Consortium, 2004; CaseMed Ontology Viewer, http://nashua.case.edu/termvisualizer). Each ontology in GO consists of GO terms associated with each other by using either the *is-a* and the *part-of* relationships. *Is-a* relationship means that the child GO term is a subclass of its parent. In the current version of GO, the *part-of* relationship means that the child is necessarily a part of its parent. That is, whenever the child GO term is assigned to a protein, the parent GO term is also assigned to that protein. As the existence of child terms always require the existence of parent terms for a protein, this situation is called the *True Path rule*. According to the True Path rule, if a protein is assigned a GO term *A*, all the GO terms on the paths from the GO term *A* to the root GO term *R*, are implicitly assigned to the protein.

Next, we apply the true path rule, and assume that a protein is indirectly annotated with all ancestor terms of its direct GO annotations. Having prepared the datasets, we ran our algorithms using correlation mining (CM) as well as the probabilistic suffix tree (PST) on the datasets. We also compared CM and PST with other known techniques, namely, neighbor counting (Schwikowski *et al*., 2000) (NC), chi-square (Hishigaki *et al*., 2001) (CHI), Markov Random Fields (Deng *et al*., 2002) (MRF). For comparison, we implemented NC and CHI techniques. For MRF comparisons, we directly used the input and prediction datasets of (Deng *et al*., 2002). In NC and CHI experiments, we used only the direct interactions of proteins (i.e., first level neighbors) since Deng *et al*. (2002) shows that using distant neighbors reduce the accuracy of CHI and NC techniques.

By applying any of the above techniques, we obtain a prediction set of GO terms. For the predicted GO terms at the deeper levels of GO hierarchy, if a parent GO term is missing in the predictions, we either add the parent term to the prediction set or remove the

GO term with a missing parent whichever requires minimum additions or deletions.

We evaluate the prediction accuracy of each technique (e.g., CM) in a k-fold cross-validation experiment. We randomly divide a protein interaction network into k clusters and use k-1 clusters as training data to annotate the excluded cluster whose annotations are marked as unknown. We repeat the same procedure many times until the accuracy of the system converges. The value of k does not significantly affect the performance of CM, NC and CHI techniques (note that results of MRF is already known) for $k \geq 5$. We chose $k = 10$, namely 10-fold cross validation to evaluate CM, NC and CHI techniques. On the other hand, our random walk algorithm for PST never visits a neighbor of a protein marked as unknown since we do not allow gaps in annotation sequences. As a result, using a small k value significantly influences the accuracy of PST due to having a disjoint training interaction network by excluding too many proteins. Therefore, in experiments, we used a larger k value, i.e., $k = 50$ to evaluate the PST technique.

Since we make experiments on already-annotated proteins, we can measure the precision and recall values of the annotation predictions. Let R be the set of (known) annotations of protein P and Q be the set of annotation predictions. Then, we define precision and recall as:

$$\text{Precision } (Q,R) = |Q \cap R|/|Q| \text{ and Recall } (Q,\ R)\ = |Q \cap R|/|R|$$

To achieve high accuracy in a prediction, the technique should have high precision and recall values. Usually there is a tradeoff between having high precision and high recall. Thus, to evaluate predictions of different techniques, we use the F-value of the prediction instead of its precision and recall. F-value is defined (Shaw *et al*., 1997) as the harmonic mean of precision and recall of a prediction set:

$$\text{F-value}(Q,R) = \frac{2 * \text{Precision}(Q,R) * \text{Recall}(Q,R)}{\text{Precision}(Q,R) + \text{Recall}(Q,R)}$$

After running one of the five techniques on a dataset, we obtain scores for all GO terms (or other annotation types). We can then obtain a prediction set by either picking the GO terms with scores above a given threshold or picking top k GO terms (with top scores). Since we compare multiple techniques, and using a threshold is not applicable due to the varying score distributions (i.e., different min, max, average scores etc...) of techniques, instead, we use the following two methods for selecting the value of k for top k cutoff in an experiment:

(i) For a given k value, we compute the average of the F-values corresponding to the top k predictions of each protein. We name this average as the ''Average F-value with Global Cutoff'' (AGC). Then we find the maximum of the AGCs (i.e., maxAGC) corresponding to a k value between 1 and the number of GO terms, to indicate the accuracy of the technique.

(ii) For each protein, we find the k value that produces the maximum F-value for the top k predictions of the protein. We name this value as ''Maximum F-value with Local Cutoff'' (MLC). Then, we average all the MLCs (i.e., avgMLC) corresponding to all proteins in order to indicate the accuracy of a technique.

	PST	CM	MRF	NC	CHI
BIO (1128)	784 (69.5%)	687 (60.9%)	685 (60.7%)	564 (50%)	503 (44.6%)
LOC (1133)	879 (77.6%)	835 (73.7%)	843 (74.4%)	655 (57.8%)	785 (69.3%)
ROLE (1398)	759 (54.3%)	734 (52.5%)	743 (53.2%)	629 (45.0%)	734 (52.5%)
AVG (1220)	807 (66.2%)	752 (61.6%)	757 (62.1%)	616 (50.5%)	674 (55.3%)

Fig. 8. Comparison of techniques by the number of proteins where a technique produces the maximum (or equal to some) MLC.

Tech.	avgMLC	Prec.	Rec.
PST	72.7%	70.1%	91.6%
CM	71.7%	69.2%	92.4%
MRF	70.7%	69.0%	90.3%
NC	68.4%	66.6%	91.5%
CHI	65.2%	62.1%	91.4%

Fig. 9. Comparison of techniques by avgMLCs over all proteins.

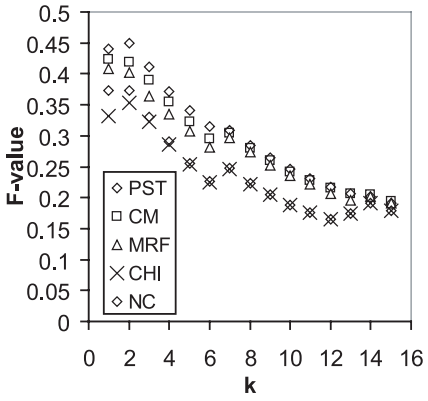


Fig. 10. AGC versus k in the top-k prediction experiments.

4.1 Comparison of techniques

In this experiment, we compare protein annotation prediction performances of five techniques, namely, correlation mining (CM), probabilistic suffix tree (PST), Markov random fields (MRF), neighbor counting (NC) and chi-square (CHI). For each technique, we compute the MLC value of each protein, and count the number of proteins where the technique produces the best (or equal to some) MLC, in comparison with other techniques (see Figure 8). We also compute the avgMLCs over all proteins (see Figure 9). In Figure 10, we plot the AGC values versus k that we compute in top-k prediction experiments.

We compare the techniques CM, PST, MRF, NC and CHI using the DENG dataset. This dataset contains three annotation classes, namely, biochemical function (BIO), cellular role (ROLE) and sub-cellular location (LOC) annotations (See Figures 8 and 9). We plot the AGC values (Figure 10) for only biochemical function annotations since the results are similar for other annotation classes.

Our results show that prediction accuracies of techniques are in the following decreasing order: PST, CM, MRF, NC and CHI. PST technique annotates 6.6%, 31% and 19.7% more proteins accurately as compared to MRF, NC and CHI techniques, respectively. CM technique annotates 22.1% and 11.6% more proteins accurately as compared to NC and CHI techniques, respectively, and 0.7% less

Sub-ontology	avgMLC
BP	63.1%
MF	60.2%
CC	75.3%

Fig. 11. avgMLCs obtained in BIND datasets using CM technique.

proteins accurately as compared to MRF technique. However, CM technique produces 1.4%, 4.8% and 10% better avgMLC values than MRF, NC and CHI techniques respectively. Comparing the avgMLCs, the PST technique gives the best results, and produces 2.8%, 6.3% and 11.5% better predictions than the MRF, NC and CHI techniques, respectively. In Figure 10 we show that the AGC difference between the techniques increases when we reduce the value of k in top-k prediction experiments. The decreasing accuracy order PST>CM> MRF>NC>CHI remains in the AGC comparison. Highest AGC values in experiments (i.e., maxAGC) is obtained for $k = 2$ (i.e., top 2 predictions).

4.2 Comparison of sub-ontologies

In this experiment, we compare different GO sub-ontologies in terms of prediction accuracies of the annotations. The different ontologies used are biological process (BP), molecular function (MF) and cellular component (CC). In Figure 11, we list the average MLCs obtained in BIND and GRID datasets using the PST technique on different sub-ontologies. Prediction results show that real scores clearly perform better than random function assignments validating the correctness of our approach.

In Figure 12, we show AGCs of different GRID dataset sub-ontologies computed in top-k prediction experiments. Among the three GO sub-ontologies, we obtain the highest accuracy predictions using the cellular component sub-ontology (in terms of AGCs for k<15 in Figure 12, and avgMLC values in Figure 11). We explain this observation as follows. Physical protein interactions occur in the same cellular component, and protein interaction partners are usually annotated with the same cellular component annotation. Therefore, GO terms belonging to the cellular component sub-ontology are usually highly correlated with themselves. As a result, to predict the annotation of a protein P, choosing highly correlated GO terms of P's interaction partners is equal to transferring most frequent GO terms of P's interaction partners. However, results of BP and MF are close (in terms of the avgMLCs) and the distribution of BP and MF annotations over a protein interaction network is too complex to have an explanation.

4.3 Comparison of Datasets

In this experiment, we compare prediction performances of different datasets (i.e., BIND, GRID, MIPS and DENG) (See Figure 13). We compute avgMLC with the CM and the PST techniques on a given dataset.

Our results show that prediction experiments on the BIND dataset performs better than GRID and MIPS datasets for the CM technique, while GRID dataset produces the best PST predictions. This is due to the fact that GRID and MIPS datasets contain protein interaction of a single organism (i.e., yeast) while the BIND dataset is a combination of protein interaction data of several organisms. Therefore, we explain the prediction accuracy difference between BIND and GRID datasets by the additional organisms in the BIND datasets. Since the BIND dataset is a multi-organism dataset and a protein does not exist in multiple organisms, the BIND dataset is

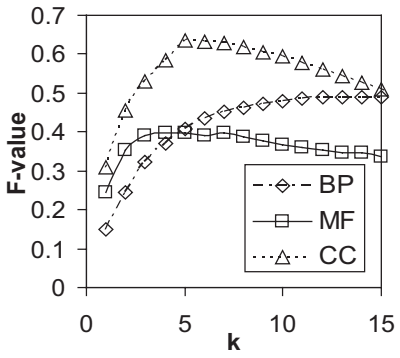


Fig. 12. CM performances of GRID sub-ontology annotations, plotting AGC versus k in top-k prediction experiments.

DATASET	CM avgMLC	PST avgMLC
BIND	66.6%	63.8%
GRID	65.8%	65.6%
MIPS	48.0%	41.6%
DENG	71.7%	72.6%

Fig. 13. Performances of data sources. Values are obtained by averaging avgMLCs in different sub-ontologies.

composed of many disjoint protein interaction networks while GRID dataset has a smaller number of disjoint portions. Hence, in PST experiments, shorter annotation sequences become more significant for the BIND dataset reducing the prediction accuracy of proteins in long protein paths. On the other hand, the CM technique does not rely on long protein paths and we are able to use the correlation information from all organisms together.

We obtained best prediction results (PST and CM) with DENG dataset. This is because the DENG dataset contains only a small number of functional annotation types (instead of GO terms) with high information content (i.e., annotation frequency).

We got the worst prediction results with the MIPS dataset. The MIPS dataset is annotated with the FunCat functional categories. FunCat is a hierarchy of functional classes combining functional categories of different types (molecular functions, cellular locations etc...) in the same hierarchy. Unrelated branches of FunCat probably reduced the overall prediction performance of this dataset.

Note that, we obtain the avgMLC values of BIND, GRID and DENG datasets by averaging the MLC values of different sub-classes (BP, MF and CC in BIND and GRID; BIO, LOC and ROLE in DENG) since different sub-classes are not related.

4.4 Effect of sampling size

In PST experiments, we repeated the same experiment with different sampling sizes using the PST technique on GRID dataset, and measured avgMLC for each sample size and the number of proteins giving better MLC values for a given sample size among all sample sizes. Our results indicate that annotation samples per protein and the number of protein samples do not change the accuracy as long as the total number of annotation samples is more than a sufficient number (i.e., 300,000) (see Figure 14) which is almost 100 times the number of proteins in the dataset.

In addition to measuring the effective number of annotation samples, we measure the effective length of the annotation sequences (i.e., the

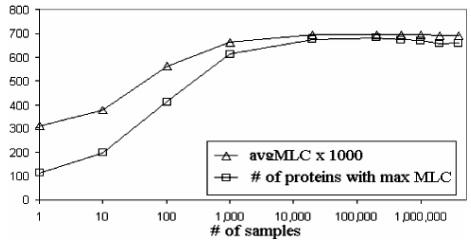


Fig. 14. Effect of sampling size on PST performance.

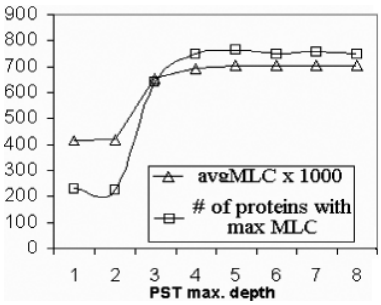


Fig. 15. Effect of PST-depth on prediction performance.

distance of effective neighbors to the target protein). We force the maximum length of annotation sequences in the PST by training the PST with a limited-length annotation sequence samples, measure the avgMLC value for each PST-depth, and compute the number of proteins giving better MLC values for a given PST-depth size among all PST-depths. We found that the PST is stabilized with the annotation sequences of length 5, and longer sequences had no improvement in the prediction accuracy (see Figure 15). However, reducing the maximum PST-depth below 5 reduces the prediction accuracy (see Figure 15).

4.5 Presentation of predictions

In this section we present our results obtained by the CM technique with the BIND dataset, since we obtained the highest avgMLC values with this dataset (See Figure 13).

The precision/recall values in Figure 16 are obtained by using the given k values and picking the top k GO terms with highest scores. The best AGC value (60%) is obtained with k = 3 where we pick the top 3 predictions.

In Figures 17 and 18, we plot the avgMLCs of proteins with the same number of interaction partners and the same number of GO term assignments, respectively. As shown in Figures 17-18, the number interactions that a protein has or the number of GO terms that a protein is assigned to do not directly influence the accuracy of the predictions.

In Figures 19 and 20, we show the correct prediction rate of individual GO terms (prediction rate = correct predictions/all predictions). As shown in Figures 19-20, GO terms with higher information content (higher number of assignments) can be predicted with better accuracy. We did not observe any relationship between information content and prediction accuracy for lower information content. GO terms with lower depth are predicted with higher accuracy in general (due to higher information content). However there are many exceptions that GO terms with higher depth are predicted with better accuracy than the GO terms with lower accuracy (see Figure 20).

k	AGC	Precision	Recall
1	50%	100%	33%
2	58%	81%	45%
3	60%	68%	53%
4	58%	58%	59%
5	57%	52%	62%

Fig. 16. Precision vs. Recall in CM experiments using the GRID BP dataset.

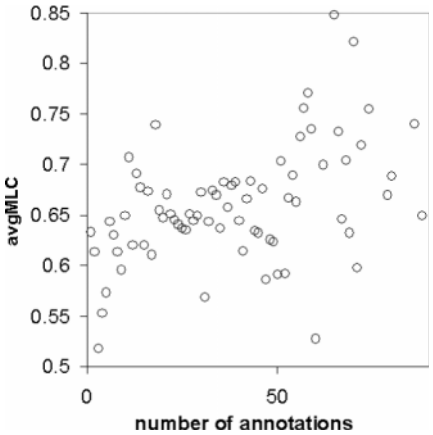


Fig. 17. Accuracy of predictions by proteins with the same number of GO term annotations.

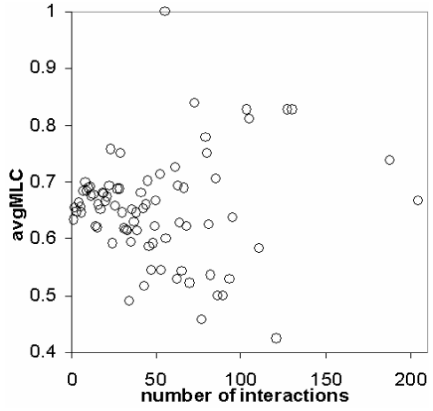


Fig. 18. Accuracy of predictions by proteins with the same number of interaction partners.

4.6 Score improvement with annotation-based correlation values

In this experiment, we observe the effects of using annotation-based correlations. When we employ annotation-based correlations to improve the prediction scores of CM technique, we obtain up to 30% improvement in individual protein MLCs. Figure 21 lists the improvements on the MLCs of the CM experiment on different datasets. Overall improvement of score update on avgMLCs is small (i.e, 0.1%–0.4). However, when annotation-based scores are employed, the effect is observed only on a set of proteins rather than all proteins, and also we observed no improvement on a large percentage of the proteins.

4.7 Effect of the correlation measure

We observe that, in GO annotations, term frequencies are non-uniform, showing some Zipf-like distribution (See Figure 22).

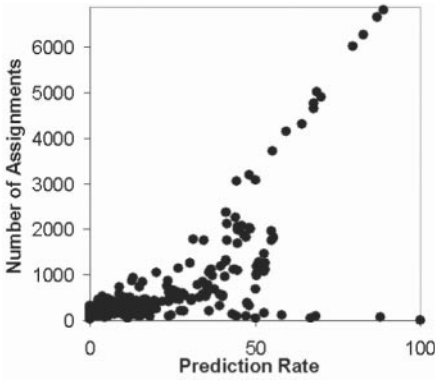


Fig. 19. Rate of correct predictions of GO terms by the number of assignments to proteins.

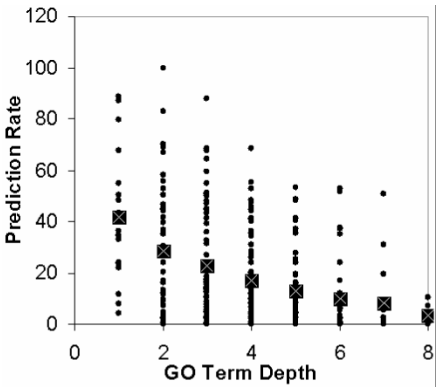


Fig. 20. Rate of correct predictions of GO terms by the depth of the GO terms in the GO hierarchy. Bigger points show the average prediction rate of GO terms with the same depth.

First, non-frequent GO terms may result in the sparseness of the data. Sparse GO terms cannot be predicted as accurately as the non-spare ones (see Figure 19), and create noise in data for prediction of non-sparse GO terms. We prevent sparseness by removing the ''uninformative GO terms'' (see section 4). Second there may exist some highly frequent GO terms, occurring in almost every protein therefore being correlated with almost every other GO term (due to a correlation measure that is proportional to co-occurrence frequency). Once we remove the uninformative GO terms, F_{11}/F_{PP} (See section 3.1.1) ratio of frequent terms reduces below 0.1%, causing no frequent item problems (He *et al*., 2004).

In this experiment, we compared the prediction performances of Cosine, Jaccard, H-measure, Support and Confidence measures by computing the avgMLCs in our datasets (See Figure 23). Cosine measure performed the best (overall) prediction results except that the H-measure performs better in the BIND dataset. The difference between the results of the Cosine and the Jaccard measures is small. H-measure is better only for the BIND dataset which is our largest dataset in terms of number of proteins and GO term annotations. In the BIND dataset, annotation frequencies become similar for frequent GO terms, and the accuracy of correlation measures using F_{11} in their formula (See Figure 3) dramatically reduces in such large datasets.

Dataset	Overall avgMLC Improvement	Maximum Individual MLC Improvement
BIND	0.4%	16.7%
GRID	0.1%	16.7%
MIPS	0.2%	17.9%
DENG	0.3%	30.0%

Fig. 21. Improvements in avgMLC and individual protein MLCs in CM experiments, by using annotation-based correlations.

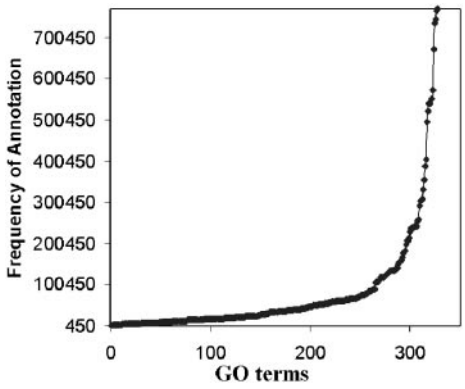


Fig. 22. Frequency of GO terms in BIND dataset.

4.8 Origin of prediction

In contrast with MRF, NC and CHI; CM and PST approaches utilize correlations between cross annotations rather than classifying proteins against a single annotation. In this experiment, we present a set of protein annotation predictions where CM performs better by utilizing cross-functional information. We list some selected predictions on the DENG dataset, to compare different techniques. We eliminated PST results from the example since PST annotations employ correlation information of annotation sequences; and due to space restrictions. Function descriptions and the full list can be found in the supplemental data available online (http://kirac.case.edu/PROTAN).

For selected proteins, Figure 24 shows top 5 predictions of different techniques and the origin of CM prediction scores assigned to the given predictions. As seen in Figure 24, in function predictions where the protein has no interaction partners with the same function annotation (e.g., YPT31 and PHO85), the whole prediction comes from cross-functional information, and other techniques fail to make an accurate prediction. Also, there are some cases (e.g., ISY1, SNF7 and NRG1) where the correct annotation of a protein is not frequent among its interaction partners, and the CM technique employs cross-functional information to increase the rank of correct predictions.

5 RELATED WORK

Related work in protein function prediction is listed briefly. Troyanskaya *et al*. (2003) builds a Bayesian Network based on the probabilities that a gene is functionally related to another to predict functional relationship between genes. Samanta and Liang (2003) puts forward that two proteins have similar functionality if they interact with a similar set of proteins, and compares shared interaction partners of two proteins. Schwikowski *et al*. (2000) counts the function annotations of proteins that interact with a non-annotated protein P in a protein interaction network and annotate P with the most frequent function annotation. Hishigaki *et al*. (2001) employs Chi-square technique on function frequencies of interaction partners

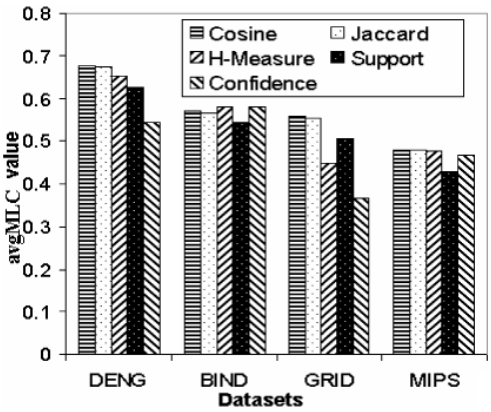


Fig. 23. Effect of difference correlation measures.

Prot.	Func. Ann.	Tech.	Top 5 Predictions	Corr. from Same	Correlation from Cross Func.
YPT31	17 21	CM	17 21 19 9 22	0%	100%
		MRF	15 21 56 42 14	-	-
		NC, CHI	18 15 30 43 16	-	-
PHO85	39 53	CM	39 53 45 34 37	0%	100%
		MRF	45 17 22 39 10	-	-
		NC, CHI	12 45 34 15 30	-	-
SNF7	50	CM	50 21 1 11 10	93.8%	6.17%
		MRF	11 21 1 50 53	-	-
		NC, CHI	1 21 11 50 15	-	-
NRG1	14 52	CM	52 14 53 22 3	63.7%	36.3%
		MRF	52 21 53 22 42	-	-
		NC, CHI	22 52 15 30 43	-	-
ISY1	42 49	CM	49 42 21 30 46	51.7%	48.3%
		MRF	21 53 52 49 42	-	-
		NC, CHI	49 15 30 43 16	-	-

Fig. 24. Utilization of cross-functional information in CM technique.

of a non-annotated protein. Vazquez *et al*. (2003) changes the problem of function prediction to a global optimization problem, i.e., minimizing the number of protein interactions between protein pairs that are annotated with different functions. Deng *et al*. improves previous techniques with a probabilistic model (2002; 2004). Deng *et al*. (2002) defines a Markov Random Field model on yeast protein interaction network that takes into consideration the fraction of the functions to be assigned to the proteins. Deng *et al*. (2004) further improves the model by defining GO terms as protein functions. Nabieva *et al*. (2005) views protein functions as reservoirs and the protein interaction network as a circuit, then predicts annotations of proteins by transferring functions, with some probability, from every other protein in the protein interaction network.

6 CONCLUSION

In this paper, we proposed a novel approach to predict GO annotations of proteins. We use protein interaction networks to find correlations and probabilistic relationships between GO terms. We use cross-validation to assess the accuracy of our algorithms. We experimentally evaluated our techniques and concluded that probabilistic suffix tree and correlation mining perform the best among the known techniques in terms of accuracy of predictions. Correlation mining performs better in large datasets (i.e., high number of proteins, high number of GO terms) and PST performs better in smaller datasets (i.e., with non-GO annotations).

ACKNOWLEDGEMENTS

This research was supported in part by the NSF award DBI-0218061, a grant from the Charles B. Wang Foundation, and Microsoft equipment

REFERENCES

Asako,K. *et al*. (2005) Automatic extraction of gene/protein biological functions from biomedical text. *Bioinformatics*, **21** (7), 1227–1236.

Begleiter,R. *et al*. (2004) On Prediction Using Variable Order Markov Models. *Journal of Artificial Intelligence Research (JAIR)*, **22**, 385–421.

Bejerano,G. *et al*. (2001) Markovian domain fingerprinting: statistical segmentation of protein sequences. *Bioinformatics*, **17**, 927–934.

Durbin,R. *et al*. (1998) Biological sequence analysis: Probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge UK.

Deng,M. *et al*. (2002) Prediction of Protein Function Using Protein-protein Interaction Data. *CSB*, 197–206.

Deng,M. *et al*. (2003) Assessment of the reliability of protein-protein interactions and protein function prediction. *PSB*, 140–151.

Deng,M. *et al*. (2004) Mapping Gene Ontology to proteins based on protein-protein interaction data. *Bioinformatics*, **20**, 895–902.

Gene Ontology Consortium (2004), The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res*., **32**, D258–D261.

Breitkreutz,B.J. *et al*. (2003) The GRID: the General Repository for Interaction Datasets. *Genome Biol*., **4**, R23.

Galil,Z. and Ukkonen,E. (1995) 6th Annual Symposium on Combinatorial Pattern Matching, volume 937 of Lecture Notes in Computer Science. Springer, Berlin.

He,B. *et al*. (2004) Discovering complex matchings across web query interfaces: a correlation mining approach. *KDD*, 148–157.

Hishigaki,H. *et al*. (2001) Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast*, **18**, 523–531.

Hu,H. *et al*. (2005) Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, **21** (Suppl 1), i213–i221.

Izumitani,T. *et al*. (2004) Assigning Gene Ontology Categories (GO) to Yeast Genes Using Text-Based Supervised Learning Methods. *CSB*, 503–504.

King,O. D. *et al*. (2003) Predicting gene function from patterns of annotation. *Genome Res*., **13**, 896–904.

Letovsky,S. and Kasif,S. (2003) Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, **19**, 197–204.

von Mering,C. *et al*. (2003) Genome evolution reveals biochemical networks and functional modules. *Proc. Natl Acad. Sci. USA*, **100** (26), 15428–15433.

Nabieva,E. *et al*. (2005) Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, **21** (Suppl. 1), i302–i310.

Poyatos,J. F. and Hurst,L. D. (2004) How biologically relevant are interaction-based modules in protein networks? *Genome Biol*., **5** (11), R93.

Shaw,W. M.,Jr *et al*. (1997) Performance standards and evaluations in IR test collections: Vector-space and other retrieval models. *Info. Proc. Manag*., **33** (1), 15–36.

Samanta,M.P. and Liang,S. (2003) Predicting protein functions from redundancies in large-scale protein interaction networks. *Proc. Natl Acad. Sci. USA*., **100** (22), 12579–83.

Schwikowski,B. *et al*. (2000) A network of protein–protein interactions in yeast. *Nat. Biotechnol*, **18**, 1257–1261.

Sharan,R. *et al*. (2005) Conserved patterns of protein interaction in multiple species. *Proc. Natl Acad. Sci. USA*., **102** (6), 1974–9.

Troyanskaya,O. G. *et al*. (2003) A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). *Proc. Natl Acad. Sci. USA*., **100** (14), 8348–8353.

Tan,P. *et al*. (2002) Selecting the right interestingness measure for association patterns. *SIGKDD*, 32–41.

Tong,A. H. Y. *et al*. (2004) Global Mapping of the Yeast Genetic Interaction Network. *Science*, 808–813.

Vazquez,A. *et al*. (2003) Global protein function prediction from protein–protein interaction networks. *Nat. Biotechnol*, **21**, 697–700.

Yang,J. and Wang,W. (2003) Cluseq: efficient and effective sequence clustering. *ICDE*, 101.

Zhou,X. *et al*. (2002) Transitive functional annotation by shortest-path analysis of gene expression data. *Proc. Natl Acad. Sci. USA*, **99** (20), 12783–8.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e271–e280
doi:10.1093/bioinformatics/btl264

A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk

Geoffrey Koh[1], Huey Fern Carol Teong[2], Marie-Véronique Clément[2], David Hsu[3] and P.S. Thiagarajan[3,*]

[1]Graduate School for Integrative Sciences and Engineering, National University of Singapore, [2]Department of Biochemistry, National University of Singapore and [3]Department of Computer Science, National University of Singapore

ABSTRACT

Parameter estimation is a critical problem in modeling biological pathways. It is difficult because of the large number of parameters to be estimated and the limited experimental data available. In this paper, we propose a decompositional approach to parameter estimation. It exploits the *structure* of a large pathway model to break it into smaller components, whose parameters can then be estimated independently. This leads to significant improvements in computational efficiency. We present our approach in the context of Hybrid Functional Petri Net modeling and evolutionary search for parameter value estimation. However, the approach can be easily extended to other modeling frameworks and is independent of the search method used. We have tested our approach on a detailed model of the Akt and MAPK pathways with two known and one hypothesized crosstalk mechanisms. The entire model contains 84 unknown parameters. Our simulation results exhibit good correlation with experimental data, and they yield positive evidence in support of the hypothesized crosstalk between the two pathways.

Contact: thiagu@comp.nus.edu.sg

1 INTRODUCTION

Computational models and methods are becoming an integral part of molecular biology. They are being used not only to identify cellular components, but also to determine how these components interact with one another. Quantitative modeling of these interactions will play an important role in understanding fundamental intra- and inter-cellular processes. In particular, quantitative modeling of the dynamics of biological pathways has drawn much attention recently (Chen *et al*., 2003; Matsuno *et al*., 2003; Ye *et al*., 2005). Our focus here is on modeling the dynamics of intra-cellular signaling pathways.

Thanks to rapid technological advances, the structures of many signaling pathways are now available. Using this information, attempts to derive system models that capture the *dynamics* of these pathways are beginning to emerge. For such attempts to be successful, several challenges must be addressed.

First, choosing a modeling framework is important, because it determines the appropriate level of abstraction at which cellular components and their interactions can be described. The choice of the modeling framework is also strongly influenced by the simulation and analysis tools that the framework offers.

Independent of the framework chosen, modeling the dynamics of a signaling pathway requires the determination of various reaction rate constants that control the biochemical reactions constituting the pathway. These rate constants are usually called model parameters. Almost always, only a few of these parameters can be determined directly through experiments. The rest must be estimated, based on experimental data, e.g., gene expression or protein concentration measurements. Unfortunately, the amount of data available is rather limited in quantity and sometimes corrupted by noise. This, combined with the large number of unknown model parameters makes the parameter estimation problem computationally difficult and sometimes intractable.

In this work, we adopt the recently introduced Hybrid Functional Petri Net (HFPN) (Matsuno *et al*., 2003) as the modeling framework and propose a *decompositional* approach to the parameter estimation problem in signaling pathway modeling. The biological application driving our study is the Akt and MAPK pathways *and* their hypothesized crosstalk mechanisms.

A key advantage of our decompositional approach is that it exploits the structure of a large pathway model to break it into smaller components, whose parameter estimation problem can then be solved independently. This leads to significant improvements in computational efficiency due to the reduction in the dimensionality of the search space and in the number of local minima. For the Akt-MAPK pathways with 84 unknown parameters, our approach produced reasonable estimates for all parameters in about 18 hours. In comparison, the common approach, which estimates all the parameters together, cannot even finish after running for 4 days.

We present our decompositional approach in the context of the HFPN modeling framework and evolutionary search (Beyer *et al*., 2002) for parameter value estimation. However, it can be easily extended to other modeling frameworks, such as simultaneous systems of differential equations, hybrid automata, etc. (Sorribas *et al*., 1988, Ye *et al*., 2005). Our approach is also independent of the specific search method used for parameter estimation. In fact, one may choose different search methods for different components, if this improves computational efficiency.

*To whom correspondence should be addressed.

We have chosen HFPN as the modeling framework, because it captures both continuous and discrete behaviors that are inherent in biological systems. Another advantage for our purposes is that the underlying graph of an HFPN model naturally captures the information flow and the dependency relations among the basic elements of a pathway. This allows us to systematically decompose a pathway model into components.

We tested our method on the Akt-MAPK pathways, based on data from 27 experiments. This pathway model has a total of 84 unknown parameters. Our method succeeded in decomposing it into 6 components, each of which has no more than 25 unknown parameters, which must then be estimated together. Our estimated parameters produced fairly good simulation results when compared with experimental data. We also used our model with its estimated parameters to check the plausibility of the hypothesized crosstalk between the protein PDK1 of the Akt pathway and the protein MEK of the MAPK pathway.

The rest of this paper is organized as follows. In Section 2, we review background information on the HFPN modeling framework and the Akt-MAPK pathways. We also provide some pointers to related work on parameter estimation techniques. In Section 3, we describe the HFPN model of the Akt-MAPK pathways and encapsulate the crosstalk hypothesis in the model. In Section 4, we present the details of our decomposition method for parameter estimation. In Section 5, we present simulation results to validate the estimated parameter values and to test the crosstalk hypothesis. In Section 6, we discuss some issues and possible improvements of our current decomposition method. Finally, in Section 7, we summarize the main results and discuss the prospects for future work.

2 THE BACKGROUND

There are many approaches to modeling biological pathways (de Jong, 2002). We first explain the modeling framework that we have chosen and then the specific signaling pathway setting in which we have carried out our parameter estimation work.

2.1 Hybrid functional petri nets

Petri nets are a fundamental model of distributed discrete event systems (Reisig, 1992). They offer an appealing visual notation which resembles the graphical notations often deployed by biologists to depict the components and their interactions in biological pathways. The Petri net model, however, has a precise semantics which fixes the meanings of the nodes and the arcs of the model as well as its *dynamics*.

A Petri net can be viewed as a bipartite graph with two kinds of nodes, usually called *places* and *transitions*. The places represent local states and the transitions represent local change-of-states. Entities called tokens are used to mark the places to specify the current distributed state of the system. The transitions, according to a *firing rule* associated with them, effect local transformations of the token distribution to model the system evolving from the current state to a new one. In the graphical representation, the places are drawn as circles, the transitions as boxes, and the tokens as small bullets placed inside the places. A standard firing rule is that if all the places pointing into the transition currently carry at least one token each, then the transition may fire. When it does so, one token is removed from each of its input places and one token is added to each of its output places. To improve modeling power, one can also

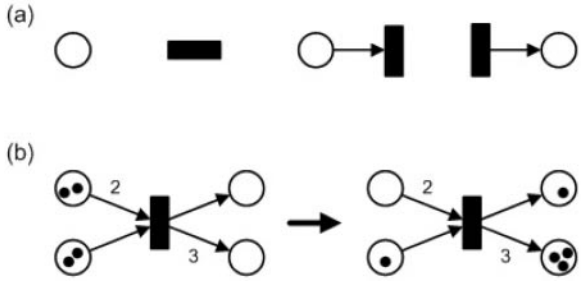


Fig. 1. (a) The basic components and connections of a Petri net model. (b) Change in markings of a Petri net due to the firing rules.

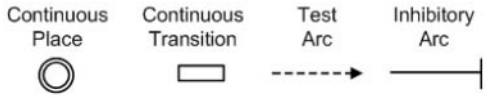


Fig. 2. Additional features of an HFPN model.

associate *weights* with the arcs so that the firing of a transition can depend on, remove and add multiple tokens from its surrounding places. This is illustrated in Figure 1(b). There are a large number of variations of this basic model, and they have been deployed in a wide variety of application domains. A recent collection of such efforts in biological settings can be found in Chen *et al*. (2003), Matsuno *et al*. (2003), Voss *et al*. (2003), Zevedei-Oancea *et al*. (2003).

The classical Petri net is a model of a discrete event system whereas a crucial aspect of biological pathways is the various bio-chemical reactions which are best specified as continuous differential equations. Indeed, both discrete and continuous features appear to be an integral part of fundamental biological processes (Lincoln *et al*., 2004). To account for this, various *hybrid* dynamic models have been proposed in the literature (Lincoln *et al*., 2004). In the setting of Petri nets, the hybrid version of interest to us is the Hybrid Functional Petri Net developed by Matsuno *et al*. (2003).

In an HFPN, places and transitions can be discrete or continuous. The marking associated with a continuous place can be a real number, which can change smoothly according to the speed assigned to the continuous transition(s) to which it serves as an input or output place. In addition, an edge can be one of three types: normal, inhibitory, or test. An inhibitory edge points from a discrete place to a transition, and it specifies that the transition is inhibited from firing whenever a token is *present* in the place. A test edge from a place to a transition specifies that the transition can only fire if a token is present in the place, *but* the firing of the transition does not change the token count on this place. See Figure 2 for HFPN features that are not present in ordinary Petri nets.

A typical biochemical equation depicting an enzyme catalyzed reaction can be written as Equation 1. In this reaction, the enzyme E binds reversibly to the substrate S, before converting it into the product P and releasing it. The parameters k_1, k_2 and k_3 are the rate constants that govern the speed of these reactions.

$$S + E \underset{k_2}{\overset{k_1}{\rightleftharpoons}} S \cdot E \overset{k_3}{\rightarrow} E + P \qquad (1)$$

The HFPN representation of such a reaction is shown in Figure 3(a). Each molecular type is represented by a continuous

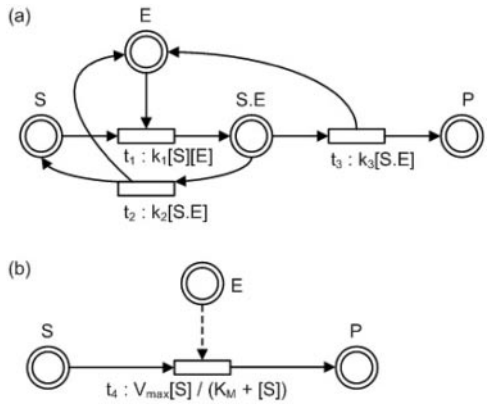


Fig. 3. (a) HFPN representation of the biochemical reaction. Assuming Michaelis-Menton kinetics, the model can be simplified into (b).

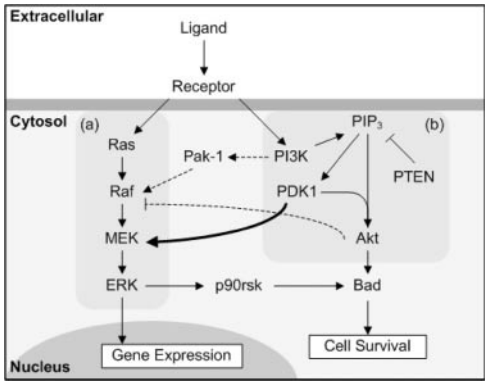


Fig. 4. Schematic of the (a) MAPK pathway, (b) Akt pathway and their crosstalk interactions. Known crosstalk interactions are marked with dashed arrows while the hypothesized interaction is marked with a bold arrow.

place, and its concentration corresponds to the marking associated with that place. The continuous transitions then represent the reactions. Each continuous transition is associated with a function which determines the speed of firing. In a biological setting, the rate of a reaction is often a function of its reactants' concentration level (denoted as [E], [S], and [E.S], respectively). In an HFPN model, this function will be attached to the continuous transition.

Assuming quasi-steady-state approximations, the HFPN model can be simplified into Figure 3(b). The function of the transition will be expressed as the Michaelis-Menton equation $V_{max}[S]/(K_M + [S])$ where

$$V_{max} = k_3[E] \quad \text{and} \quad K_M = \frac{k_2 + k_3}{k_1}$$

We have adopted the HFPN to model the Akt and the MAPK pathways and their crosstalk. Our choice of this formalism has been influenced by the fact that it serves as the front-end of the software Cell Illustrator with which the HFPN-based models can be simulated (Nagasaki *et al*., 2003). In many settings, an attractive alternative is the *hybrid automata* modeling framework (Henzinger, 1996) with its direct use of differential equations to capture the continuous dynamics. This model has an extensive theory and an emerging set of simulation, analysis, and verification tools (Lincoln *et al*., 2004). It has been used to study, for instance, the excitable behavior of cardiac cells (Ye *et al*., 2005), Delta-Notch protein signaling (Ghosh *et al*., 2001) and quorum sensing in bacteria (Alur *et al*., 2001).

2.2 The Akt pathway

The kinase Akt plays an important role in the regulation of cellular functions. Its downstream targets include kinases, transcription factors and other regulatory molecules (Khwaja, 1999). Akt has also been identified as a major factor in many types of cancer. The schematic describing the Akt pathway, its interactions with the mitogen-activated protein kinase (MAPK) pathway, and their downstream targets are shown in Figure 4.

The activation of the Akt signaling pathway is a multi-step process (Bellacosa *et al*., 1998). When ligands such as fibroblast growth factors, epidermal growth factors and insulin bind to their specific membrane receptors, the cytosolic domains of the receptors will undergo conformational changes, allowing them to act as scaffolds for certain types of proteins in the cell. Phosphoinositide 3-kinase (PI3K) is one such protein that gets recruited and as a result, its catalytic subunit will be activated. The activated PI3K will then phosphorylate the membrane phospholipid phosphatidylinositol-4,5-biphosphate (PIP_2) at the 3-OH position to phosphatidylinositol-3,4,5-triphosphate (PIP_3) and this is tightly regulated by the phosphatase and tensin homolog (PTEN), which removes the phosphate group from the same position.

PIP_3 recruits Akt and the phosphoinositide-dependent kinase-1 (PDK1) to the plasma membrane allowing the phosphorylation of Akt by PDK1. Akt is activated by a sequential phosphorylation at its threonine residue 308 (Thr^{308}) and serine residue 473 (Ser^{473}) by PDK1 and an unknown kinase (named PDK2) respectively (Nicholson *et al*., 2002). Activated Akt further phosphorylates and activates its downstream targets such as Forkhead transcription factor (FKHR) and glycogen synthase kinase 3β (GSK-3β).

Another important molecular target for Akt signaling is Bad, a protein that regulates apoptosis. Bad can bind to the anti-apoptotic proteins Bcl-2 and Bcl-XL, allowing the pro-apoptotic protein Bax to oligomerize at the mitochondria and promote the release of cytochrome c into the cytosol. This would lead to the activation of caspases and cell death. Upon phosphorylation by Akt at the Ser^{136} residue, Bad is sequestered in the cytosol by 14-3-3 proteins, thus Bcl-2 and Bcl-XL can bind to Bax, hence preventing the release of cytochrome c and inhibiting apoptosis. Constitutive Akt signaling promotes cell survival and proliferation, leading to the formation of tumors.

2.2.1 MAPK crosstalk The significance of the Akt pathway lies not only in the several cellular functions it regulates, but also in its interactions with other pathways (Heldin, 2001). The MAPK signaling cascade is one such pathway that is influenced by components of the Akt pathway.

The MAPK pathway is a highly conserved pathway that is linked to mitogenic responses and cell proliferation. It can be activated by a wide range of growth factors and hormones and it too has several target molecules. Some of the signals that activate the Akt pathway can also activate the MAPK pathway. Upon activation, the tyrosine residues of the receptor is phosphorylated, serving as docking sites for proteins such as Grb2 and SOS. The exchange factor

SOS then replaces guanosine 5′-diphosphate (GDP) on the Ras protein with guanosine 5′-triphosphate (GTP), thus activating it. Activated Ras then binds to the protein Raf, triggering a wave of downstream phosphorylation where Raf activates the MAPK kinase (MEK) which in turn activates p44/42 MAPK (ERK).

Recent studies show that the Akt pathway can regulate the MAPK pathway by competitively phosphorylating Raf at Ser^{259} (Moelling *et al*., 2002), preventing further activation. However, this Akt-MAPK regulation is not entirely inhibitory. PI3K has been shown to activate Raf via the intermediate protein PAK-1 (Chaudhary *et al*., 2000). Recently, Sato *et al*. (2004) have shown that PDK1 is involved in the activation of MEK. Moreover, our data also shows that the repression of PDK1 gene expression using small interference RNA (siRNA) leads to a decrease in activated MEK and ERK in a prostate cancer model (Teong *et al*., 2006). Thus these support our hypothesis that PDK1 could be involved in the activation of MEK by phosphorylating it, as indicated by a bold arrow in Figure 4.

2.3 Parameter estimation

Various techniques based on global optimization have been proposed for estimating the parameters of pathway models (see, e.g., (Kikuchi *et al*., 2003; Moles *et al*., 2003). However, these techniques, which usually estimate all the model parameters together, do not scale up well for large pathway models with many parameters, due to the high dimensionality of the search space and the presence of many local minima.

For larger pathway models, it is natural to try to decompose it into small, independent components and estimate the parameters for each component separately, thus reducing the computational complexity. The general idea of model decomposition for parameter estimation has been successfully applied in many domains, e.g., Bayesian model learning (Neapolitan, 2003), geometric curve fitting (Jiang *et al*., 2005), control of large dynamical systems (Williams *et al*., 1998), etc.

In related work on Akt and MAPK pathways (Hatakeyama *et al*., 2003), a simplified model based on simultaneous differential equations is proposed. The model has about 30 unknown rate parameters, which are estimated with an in-house genetic algorithm. There is no report of computation time required. It is also not clear how much experimental data was used and how the estimated parameters were validated. To ease the computational burden, in subsequent work (Kimura *et al*., 2004), the model is decomposed *manually* based on the observation that parameters in upstream components of enzyme catalyzed reactions can be estimated independent of parameters in downstream components, if there are no feedback loops connecting them. Our decompositional approach uses a similar observation, but is more general, as it is not restricted to enzyme catalyzed reactions. It is also fully automatic.

3 THE HFPN MODEL OF THE AKT-MAPK PATHWAYS

We have modeled the Akt pathway and the MAPK pathway as well as the hypothesized crosstalk as an HFPN model. The full structure of the model is shown in Figure 5. The parameters associated with the transitions and the initial protein concentration levels are shown in Table 1 and Table 2. In Table 1, the four parameters whose values have been taken from literature are marked with a^* while the remaining parameters have values that have been estimated by our technique. The sources of information of the four known parameters can be found in: http://www.comp.nus.edu.sg/~rpsysbio/ismb2006.

The model can be viewed as separate modules interacting with one another via shared nodes. Figure 5(a) models the reactions that take place when the receptors are activated by external signals (indicated by the place 'Serum'). It also includes the reactions of the Akt pathway. A point to note is that under prolonged activation, the cells become desensitized to the signals and respond less to it. We have modeled this phenomena as receptor internalization (Reaction 2).

The MAPK pathway is depicted by Figures 5(b),(c) and (d). After activation by the receptors, Ras will catalyze the phosphorylation of Raf. Phosphorylated Raf, denoted as 'Rafp', will then phosphorylate MEK at two sites, forming the doubly phosphorylated MEKpp (Figure 5(c)). Finally, ERK will be phosphorylated by MEKpp in the same manner, as shown in Figure 5(d).

In our model, there are three possible paths for crosstalk interactions: Active PI3K can upregulate the phosphorylation of Raf via PAK1(Reactions 48 and 18). Akt can inhibit Raf activity (Reaction 20), and PDK1 can affect MEK phosphorylation (Reactions 22 and 25) by the hypothesized interactions. The protein PP2A is an ubiquitous phosphatase which reverses the action of several kinases in our pathway. Hence it is not considered as exclusively belonging to the Akt pathway or the MAPK pathway.

Figure 5(f) models the activity of the Bcl-2 family members, which include the proteins Bcl-2, Bad and Bax. As mentioned in the previous section, these proteins play important roles in regulating apoptosis. It has also been shown that ERK can regulate Bad phosphorylation through the activation of the protein P90RSK, shown in Figure 5(e).

Our model consists of 44 places connected to 51 transitions. For most of the transitions, their dynamics are driven by Michaelis-Menton equations (e.g. Reactions 4, 5, 6). The rest are either association/dissociation reactions (e.g. Reactions 42, 43) or synthesis/degradation reactions (e.g. Reactions 46, 47) whose rates are governed by the mass action laws. Each of these reactions have one or two parameters associated with them.

4 PARAMETER ESTIMATION

Parameter estimation can be viewed as an optimization problem with differential-algebraic constraints. The input to the problem consists of the values of state variables at selected discrete time points. In the present setting, these are steady-state or time-series measurements of protein concentration levels. The problem is to determine the values of the m parameters $\mathbf{p} \in \mathbb{R}_+^m$ and all the unknown protein concentration levels such that they minimize the following objective function:

$$J(\mathbf{p}) = \sum_{i,j,t_e} \sqrt{\frac{\left(x_{ij}(t_e,\mathbf{p}) - x_{ij}^{\exp}(t_e)\right)^2}{w_{ij}^2}} \tag{2}$$

subject to

$$\dot{\mathbf{x}} = f(\mathbf{x},t) \tag{3}$$

$$h(\mathbf{x},t) \geq 0 \tag{4}$$

$$\mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U \tag{5}$$

Here f is the set of differential constraints describing the system dynamics. h is the state constraints for the variables $x \in \mathbf{x}$ for

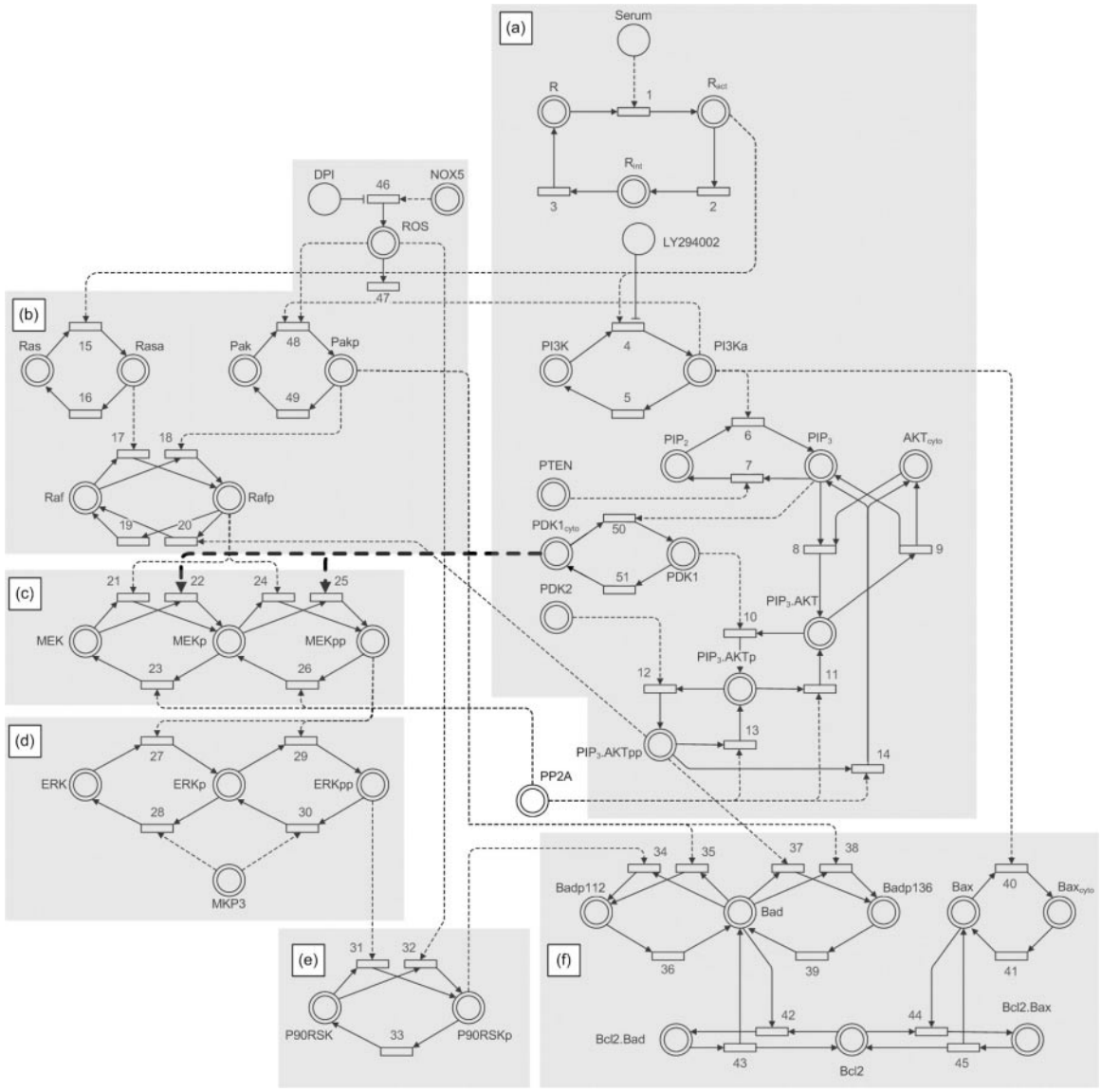


Fig. 5. HFPN model of the Akt and MAPK pathways. The hypothesized crosstalk interaction is emphasized by bold test arcs. The members of downstream components should include the places of the upstream components which they are directly linked from. However to reduce clutter, we show the components as separate modules.

all $t \in [0,T]$. $\mathbf{p}^L$ and $\mathbf{p}^U$ are the lower and upper bound constraints on the parameters $\mathbf{p}$. The time points $t_e \in T_e \subset [0,T]$ describe the set of time instances where experimental data are available. The expression $x_{ij}(t_e, \mathbf{p})$ is the model predicted value of the variable x_i in experiment j at time t_e using parameters $\mathbf{p}$ while $x_{ij}^{\exp}(t_e)$ is the experimental measurement of the same variable. w_{ij} is the weight that is used to normalize the contributions of each term to the objective function. This value is usually taken to be the maximum value of x_i in experiment j.

A typical parameter estimation algorithm starts by randomly choosing parameter values from the search space $\mathbb{R}_+^m$. It uses (3) to simulate the system according to the chosen parameters and then uses (2) to compare the results with the input data. The results of this comparison provide the information to improve the parameter values through gradient descent or stochastic search (Pardalos *et al*., 2002). This process repeats until a better solution can no longer be found or a pre-specified maximum number of iterations is reached.

Our optimization problem is highly non-linear and we will use the evolution strategies algorithm to solve this problem. This algorithm keeps a working set of μ candidate solutions. Each solution consists of a vector of parameter values. In each iteration, it randomly selects two parent solutions from the working set and generates a new one, possibly by interpolating the values of parent vectors. Thereafter, it alters the values slightly and scores the new solution by simulating the model according to these parameter values and applying the objective function. A number of such solutions are generated and from the combined set, the best scoring μ

Table 1. Rate reactions and their associated parameters. The Michaelis-Menton constants (K_M) are given in nM. The maximal rate constants (V) are expressed in nM.s^{-1}. The first order and second order rate constants (k) are given in s^{-1} and nM^{-1}.s^{-1} respectively

No	Rate Equation	Parameter	
1	k_1 [R]	$k_1 = 0.01$	
2	k_2 [R_{act}]	$k_2 = 0.002$	
3	k_3 [R_{int}]	$k_3 = 0.001$	
4	k_4 [R_{act}][PI3K]/(Km_4 + [PI3K])	$k_4 = 0.3$	$Km_4 = 78$
5	V_5 [PI3Ka]/(Km_5 + [PI3Ka])	$V_5 = 46.2$	$Km_5 = 117$
6	k_6 [PI3Ka][PIP_2]/(Km_6 + [PIP_2])	$k_6 = 0.05$	$Km_6 = 6170$
7	k_7 [PTEN][PIP_3]/(Km_7 + [PIP_3])	$k_7 = 5.5$	$Km_7 = 80.9$
8	k_8 [PIP_3][AKT_{cyto}]	$k_8 = 0.045$	
9	k_9 [PIP_3.AKT]	$k_9 = 0.089$	
10	k_{10} [PDK1][PIP_3.AKT]/(Km_{10} + [PIP_3.AKT])	$k_{10} = 20$	$Km_{10} = 80000^*$
11	k_{11} [PP2A][PIP_3.AKTp]/(Km_{11} + [PIP_3.AKTp])	$k_{11} = 0.037$	$Km_{11} = 8800$
12	k_{12} [PDK2][PIP_3.AKTp]/(Km_{12} + [PIP_3.AKTp])	$k_{12} = 20$	$Km_{12} = 80000^*$
13	k_{13} [PP2A][PIP_3.AKTpp]/(Km_{13} + [PIP_3.AKTpp])	$k_{13} = 0.04$	$Km_{13} = 48000$
14	k_{14} [PP2A][PIP_3.AKTpp]/(Km_{14} + [PIP_3.AKTpp])	$k_{14} = 0.163$	$Km_{14} = 48000$
15	k_{15} [R_{act}][Ras]/(Km_{15} + [Ras])	$k_{15} = 50$	$Km_{15} = 20000$
16	V_{16} [Rasa]/(Km_{16} + [Rasa])	$V_{16} = 15000$	$Km_{16} = 7260$
17	k_{17} [Rasa][Raf]/(Km_{17} + [Raf])	$k_{17} = 0.09$	$Km_{17} = 50$
18	k_{18} [Pakp][Raf]/(Km_{18} + [Raf])	$k_{18} = 0.183$	$Km_{18} = 500$
19	V_{19} [Rafp]/(Km_{19} + [Rafp])	$V_{19} = 78$	$Km_{19} = 30$
20	k_{20} [PIP_3.AKTpp][Rafp]/(Km_{20} + [Rafp])	$k_{20} = 0.1$	$Km_{20} = 13.2$
21	k_{21} [Rafp][MEK]/(Km_{21} + [MEK])	$k_{21} = 5.6$	$Km_{21} = 7200$
22	k_{22} [$PDK1_{cyto}$][MEK]/(Km_{22} + [MEK])	$k_{22} = 0.04$	$Km_{22} = 2600$
23	k_{23} [PP2A][MEKp]/(Km_{23} + [MEKp])	$k_{23} = 0.45$	$Km_{23} = 1250$
24	k_{24} [Rafp][MEKp]/(Km_{24} + [MEKp])	$k_{24} = 5.17$	$Km_{24} = 24500$
25	k_{25} [$PDK1_{cyto}$][MEKp]/(Km_{25} + [MEKp])	$k_{25} = 0.05$	$Km_{25} = 2150$
26	k_{26} [PP2A][MEKpp]/(Km_{26} + [MEKpp])	$k_{26} = 0.4$	$Km_{26} = 4316$
27	k_{27} [MEKpp][ERK]/(Km_{27} + [ERK])	$k_{27} = 0.089$	$Km_{27} = 52000$
28	k_{28} [MKP3][ERKp]/(Km_{28} + [ERKp])	$k_{28} = 30$	$Km_{28} = 160$
29	k_{29} [MEKpp][ERKp]/(Km_{29} + [ERKp])	$k_{29} = 0.0308$	$Km_{29} = 55000$
30	k_{30} [MKP3][ERKpp]/(Km_{30} + [ERKpp])	$k_{30} = 32$	$Km_{30} = 60$
31	k_{31} [ERKpp][P90RSK]/(Km_{31} + [P90RSK])	$k_{31} = 0.0017$	$Km_{31} = 97.6$
32	k_{32} [ROS][P90RSK]/(Km_{32} + [P90RSK])	$k_{32} = 0.76$	$Km_{32} = 181$
33	V_{33} [P90RSKp]/(Km_{33} + [P90RSKp])	$V_{33} = 468$	$Km_{33} = 2.8$
34	k_{34} [P90RSKp][Bad]/(Km_{34} + [Bad])	$k_{34} = 0.798$	$Km_{34} = 10$
35	k_{35} [Pakp][Bad]/(Km_{35} + [Bad])	$k_{35} = 0.04$	$Km_{35} = 30000$
36	V_{36} [Badp112]/(Km_{36} + [Badp112])	$V_{36} = 821$	$Km_{36} = 43300$
37	k_{37} [PIP_3.AKTpp][Bad]/(Km_{37} + [Bad])	$k_{37} = 0.397$	$Km_{37} = 20700$
38	k_{38} [Pakp][Bad]/(Km_{38} + [Bad])	$k_{38} = 0.04$	$Km_{38} = 30000$
39	V_{39} [Badp136]/(Km_{39} + [Badp136])	$V_{39} = 821$	$Km_{39} = 43300$
40	k_{40} [PI3Kp][Bax]/(Km_{40} + [Bax])	$k_{40} = 0.0659$	$Km_{40} = 42000$
41	k_{41} [Bax_{cyto}]	$k_{41} = 0.0148$	
42	k_{42} [Bad][Bcl2]	$k_{42} = 0.0561$	
43	k_{43} [Bcl2.Bad]	$k_{43} = 0.0624$	
44	k_{44} [Bax][Bcl2]	$k_{44} = 0.002^*$	
45	k_{45} [Bcl2.Bax]	$k_{45} = 0.02^*$	
46	k_{46} [NOX5]	$k_{46} = 0.00038$	
47	k_{47} [ROS]	$k_{47} = 0.0155$	
48	k_{48} [ROS][PI3Kp][Pak]/(Km_{48} + [Pak])	$k_{48} = 0.14$	$Km_{48} = 482$
49	V_{49} [Pakp]/(Km_{49} + [Pakp])	$V_{49} = 83000$	$Km_{49} = 29100$
50	k_{50} [PIP_3][$PDK1_{cyto}$]	$k_{50} = 0.0007$	
51	k_{51} [PDK1]	$k_{51} = 0.98$	

solutions are selected for the next iteration. This carries on for a certain number of iterations, or until no better solutions can be obtained (Beyer *et al*., 2002).

Common approaches to parameter estimation try to estimate all the parameters together. This leads to a high-dimensional search space and hence to very high computational complexity. The key feature of our approach is to exploit the *structure* of a pathway, to break down the parameter estimation problem into a series of smaller problems. The structure of a pathway model determines the causal links and dependencies between the system variables. We use

Table 2. Initial concentration of the cellular components

Place	Concentration (nM)
R	80
PI3K	100
PIP_2	7000
PTEN	0.1
AKT_{cyto}	200
$PDK1_{cyto}$	1000
PDK2	3
PP2A	150
RAS	18900
RAF	66.4
MEK	36500
ERK	34900
MKP3	2.4
P90RSK	5
BCL2	100
BAD	100
BAX	100
NOX5	2000
ROS	200
PAK	500

this dependency relationship to extract pathway components that can be handled independently.

This decompositional approach can be applied to different modeling frameworks. Furthermore, it is independent of the specific search method (Beyer *et al*., 2002, Kirkpatrick *et al*., 1983, Moles *et al*., 2003) used for parameter optimization. Here we present our method in the context of the HFPN model combined with evolutionary search (Beyer *et al*., 2002).

4.1 Pathway decomposition

The goal of pathway decomposition is to extract components from the pathway model whose parameter estimation problems can be solved independently. A component is an executable subgraph of the HFPN model. By an ''executable'' subgraph we mean a subgraph that can be simulated as a model by itself, assuming we have the values for the parameters and initial conditions relative to the nodes in this subgraph. It is not difficult to see that a component is executable if and only if its set of nodes (places and transitions) are *closed* relative to the full model in the following sense. If a place node is present in the component, all its incoming transitions must also be present in the component. Furthermore, all the transitions to which the place is connected via normal arcs must also be present in the component. This is so since the reactions associated with these transitions are precisely those that determine the concentration levels of the protein associated with the place. If a place is connected to a transition via an inhibitory arc or a test arc then the reaction associated with the transition does not affect the concentration level of the place in any way. By similar reasoning, if a transition is present in a component, all its input places must also be in the component.

Since there are many components (the whole model itself is a component), we must choose them in a systematic fashion so as to help decompose the parameter estimation problem. To do so, we first color the nodes of the model. We then compute a component using the criterion to be described below. We then solve the parameter estimation problem for this component. This is followed by updating the colors of some of the nodes. We then proceed to compute a second component and so on.

As a first step, we assign colors to each place. We assume we have experimental data that has been produced by K experiments conducted under different conditions. With each place we associate a K-dimensional color vector. Suppose the jth experiment produces time series values and/or the steady state concentration level of the protein associated with the place p. Then the jth component of the color vector of p is set to be grey. Otherwise it is fixed to be white. If one or more components of the color vector of a place is grey then the color of the place is defined to be grey. Otherwise it is defined to be white.

The transitions are initially colored as follows. Due to the nature of the reactions being represented by the transitions, each transition can have one or two rate parameters associated with it. If all the parameters associated with a transition are known, then the transition is colored black. If none of the parameters associated with a transition are known then it is colored white. If one but not both the parameters associated with a transition are known, then it is colored grey.

To see how we choose our first component, let $H = (P, T, h, C, a)$ be an HFPN where

- P is the set of places,
- T is the set of transitions,
- $h : P \cup T \rightarrow$ {discrete, continuous} labels the places and the transitions as being discrete or continuous.
- $C \subseteq \{(P \times T) \cup (T \times P)\}$ is the set of arcs.
- $a : C \rightarrow$ {normal, test, inhibitory} labels arcs as being normal, test or inhibitory.

Let p be a grey colored place. Then a particular component containing p-let us denote this component as $comp(p)$- is defined to be the least set of nodes of H satisfying the following conditions.

(C1) $p \in comp(p)$.
(C2) Suppose $x \in comp(p) \cap P$ and x is colored grey or white and (y, x) is an arc in H then y is also in $comp(p)$.
(C3) Suppose $x \in comp(p) \cap T$ and (y, x) is an arc in H then y is also in $comp(p)$.
(C4) Suppose $x \in comp(p) \cap P$, x is colored grey or white, (x, y) is an arc in H and (x, y) is a normal arc, then y is also in $comp(p)$

It is easy to see that $comp(p)$ is indeed a component. Now from among all components $\{comp(p)\}$ where p ranges over the set of grey places, we choose the one which has a minimum number of white/grey transitions while maximizing the number of grey places. Thus choosing a component is a non-trivial task. A number of strategies can be adapted to ease this task but we will not address them here. In any case, this part of the procedure consumes only a small fraction of the overall time needed.

Suppose we have chosen the component $comp(p_0)$ corresponding to the grey place as the best according to our criterion. In the model shown in Figure 5, p_0 is PIP_3. AKTpp and the component it generates is highlighted in Figure 5(a). We now apply the evolutionary search procedure to $comp(p_0)$ using a reserved fraction of the

experimental data that provides values for the concentration levels of the proteins associated with the grey places in $comp(p_0)$. This search involves simulating the component several times, adjusting the parameters in each iteration to get a better result according to the specifics of the evolutionary procedure which we will not get into detail here. During this phase, the estimation procedure might get stuck in a local minima while significantly differing from the reported data. Such situations are dealt with in two possible ways. The first consists of using biological intuition to bump the current estimated parameter values out of the local minima trap. The second is to examine the concerned experimental data and discard it as being too noisy or as being improperly conditioned to be reliable.

At the end of this first phase, all the parameters associated with the transitions in $comp(p_0)$ would have been estimated. We now color all the nodes in $comp(p_0)$ as black.

We now choose a suitable grey place p_1 in $P - comp(p_0)$ and compute $comp(p_1)$ and repeat the above process. It is worth noting that when computing $comp(p_1)$, black-colored places of $comp(p_0)$ will form the boundary nodes of this new component. This is because, according to our back-tracing procedure for computing components, the input transitions of a black-colored place will *not* be included in the new component. This implies that each new component will include only a small portion of components that have already been computed. This leads to reduced computation time for each of our parameter estimation task.

After a finite number of iterations all the parameters would have been estimated and all the nodes would have been colored black. For the model shown in Figure 5, the sequence of components chosen is (a), (b), (c), (d), (e) and (f).

We then check the accuracy of the estimated parameters by simulating the model using the estimated parameters and the fraction of the experimental data that has been reserved for this purpose.

5 SIMULATION AND RESULTS

We now describe our results on parameter estimation (Section 5.1) and on the hypothesized Akt-MAPK crosstalk mechanisms (Section 5.2).

5.1 Parameter estimation

Using the method described in the previous section, we performed parameter estimation for the Akt-MAPK pathways. The data for the estimation problem was obtained from 27 experiments, of which 10 provided time-series data and 17 provided steady-state data. The 27 experiments were performed under 18 different initial conditions. We used data from 23 experiments as inputs to our parameter estimation procedure and reserved data from 4 time-series experiments to validate the results of estimation. The data reserved for validation was not revealed to the estimation procedure. All the data files that we used are available for download from http://www.comp.nus.edu.sg/~rpsysbio/ismb2006.

The Akt-MAPK pathways consist of 51 reactions and a total of 88 parameters, of which 4 are known. We assume that the initial conditions for all the places are known, and this is supported partly by experimental data. For other situations where not all the initial values are known, the missing protein concentrations can be treated as parameters to be estimated as well.

We ran the estimation procedure on a Pentium 4 PC with a 2.8GHz processor and 2.5GB memory. Our procedure broke the pathways into 6 components (Figure 5(a)–(f)). The average time to estimate the parameters for each component was about 3 hours, and the total time to estimate all the parameters was about 18 hours.

For comparison, we also tried to solve the same parameter estimation problem using the global approach; in other words in which one tries to estimate all the parameters together using the evolutionary search method. On the same computational platform, after running for 4 days, the global method failed to produce a set of parameters that can produce reasonable simulation results. See Figure 6 for a comparison.

An obvious measure to assess the accuracy of the parameters and the reliability of the parameter estimation method is the deviation of the simulation results from experimental data. Figure 7 shows the simulation results of both MEK and ERK activity and Figure 8 shows the Bad phosphorylation levels for the experiment where the cells are treated with diphenyleneiodonium (DPI), a NADPH oxidase inhibitor, in the presence of serum. Due to the lack of space, we do not show the results for all the proteins here. Figure 7 shows that the match between the simulation results and experimental data is good, though not perfect. Given the limited, noisy data available and the high dimensionality of the search space, these results represent a reasonable first step. We expect that as we generate better data both in terms of quantity and quality, a better match will be obtained. Longer running time for the estimation procedure may also potentially help.

Among the results obtained, we indeed have cases in which the match between the simulation results and experimental data is not good. Figure 8 shows the simulation results and experimental data for phosphorylated Bad. There is a systematic, constant difference between them. From a static analysis of the model in Figure 5, the inhibition of the production of superoxide (ROS) by treatment with DPI will propagate downstream and we would expect the level of phosphorylated Bad at Ser^{112} to decrease (Figure 5(f)). However, the experimental data points for activated Bad in Figure 8 are consistently higher than those from the cells which have not been treated with DPI. This difference could be due to unknown reactions and needs to be further investigated.

5.2 Effects of PDK1 on MEK and ERK

A key biological motivation for this work was to test the plausibility of the hypothesized crosstalk interaction between the Akt and the MAPK pathways. Experiments show that in LNCaP, a prostate cancer cell line (Horoszewicz *et al*., 1983), transfected with PDK1 siRNA, which reduces the total PDK1 in the cell, results in a significant decrease in the phosphorylation of MEK and ERK. This suggests a possible crosstalk with PDK1 activating MEK by phosphorylation. We performed the same simulations, decreasing the levels of PDK1 (from 1000 nM to 0 nM) to mimic the knock-down of PDK1 using siRNA. Figure 9 shows the results of the simulations. ERK activity is being reduced to negligible levels after decreasing the amounts of PDK1.

However, PDK1 seems to exist in an active conformation under normal conditions (Vanhaesebroeck *et al*., 2000). With the hypothesized interaction, one would assume that in the absence of external signals, PDK1 will continuously activate the MAPK pathway, causing uncontrolled growth. This contradicts the current view that the MAPK pathway is activated by external signals.

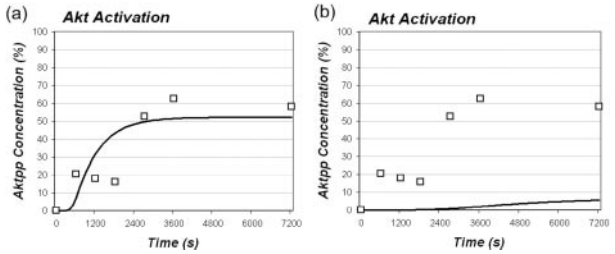


Fig. 6. Comparison of the simulation results of Akt against experimental data using the parameters estimated with (a) the decompositional method and (b) the conventional method. ('□'—experimental data points, '—' simulation profiles).

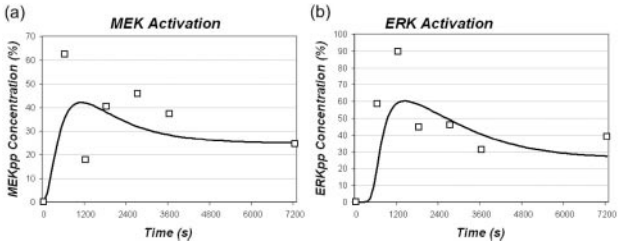


Fig. 7. Simulation profiles of (a) MEK and (b) ERK activation levels.

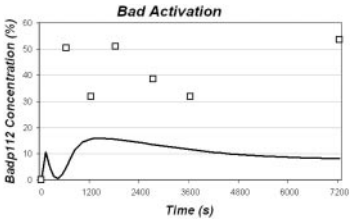


Fig. 8. Simulation profiles of Bad activation levels compared to experimental data.

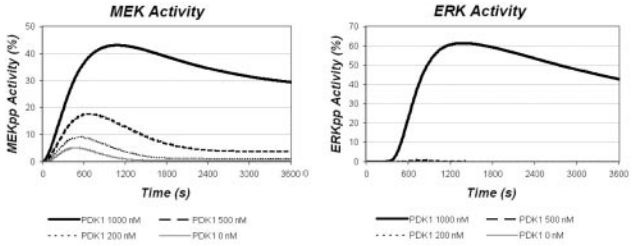


Fig. 9. Simulation of MEK and ERK activation levels with decreasing amounts of PDK1.

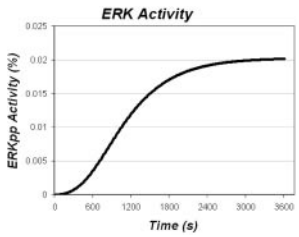


Fig. 10. ERK activation levels in the absence of serum.

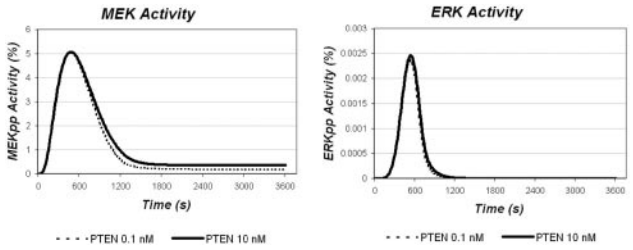


Fig. 11. MEK and ERK activation levels without PDK1 crosstalk.

However, simulations show that under serum starved conditions, PDK1 does activate ERK, but its activation is maintained at a low basal level of 0.02% (Figure 10). This suggests that PDK1 may indeed be a necessary but not sufficient condition to fully activate the MAPK pathway, therefore lending support to the presence of the interaction.

We also tested the possibility of *not* having this crosstalk interaction between PDK1 and MEK by removing it from our model. Simulations of this pathway configuration revealed that the ERK activity was kept low throughout even in the presence of serum. This, we suspect, could be due to the inhibitory effect of activated Akt (Moelling *et al*., 2002). To further confirm this observation, we took into account the fact that our model was based on the LNCaP cell line which has defective PTEN due to a frameshift mutation in the *PTEN* gene. Hence we re-simulated the model (with the PDK1-MEK interaction removed) with 10 nM of PTEN. This simulation produced a similar outcome (Figure 11). However, these simulations should not be taken as a conclusive comparison as the modified configuration (without the interaction) could not fit the experimental data.

Although more experiments are needed to confirm the role of PDK1 in the regulation of MAPK activity, the above simulations suggest that the interaction is not only present but also necessary for enabling the MAPK pathway. This also seems to imply that knocking down PDK1 to reduce Akt activity may affect the proper functioning of the MAPK pathway.

6 DISCUSSION

Decomposition of a large system into smaller sub-systems is an effective way to make the parameter estimation problem manageable. The method proposed here is our first attempt at systematically decomposing dynamical models of signaling pathways. It can be improved in several ways. Currently, we decompose a pathway into components by using the dependency relations among the places and transitions of the HFPN model. For inverse problems such as parameter estimation, information from the downstream components can also possibly aid in constraining the search space for upstream components. At present, we are not taking advantage of this. It should be possible to use techniques such as constraint propagation (Tucker *et al*., 2005) to push up information from downstream components to upstream components.

Also, our decomposition method is most effective when the flow of information is one-way or when the feedback loops are short. If the pathway components are tightly coupled together or if there are

long feedback loops, our method may return the entire pathway as the first component. One possible way of dealing with this is to use splines to approximate the concentration profiles of grey places, so that they can be viewed as black places. By doing so, these places can then serve as the boundaries for the smaller components that will be generated.

7 CONCLUSION

In this work, we have built an HFPN model for the Akt and MAPK signaling pathways and investigated their hypothesized crosstalk interaction. Pathway simulation results based on our estimated model parameters exhibit good correlation with experimental data and support a new hypothesized crosstalk mechanism linking the Akt pathway to the MAPK pathway.

One main contribution of this work is a decompositional method for model parameter estimation, based on the HFPN representation. By breaking a large pathway model into smaller, independent components, the new method offers significant improvement in computational efficiency. It shows considerable potential for scaling up to large pathways with hundreds of parameters, a task too daunting for conventional methods. As described in Section 6, there are several improvements that can be made on our current decomposition method, and we are currently working on them. We also plan to extend our approach to other modeling frameworks, such as simultaneous differential equations, hybrid automata, and stochastic Petri nets. The idea is to capture the dependency relations among the pathway elements in the form of a dependency graph similar to the bipartite graph that underlies an HFPN model. On the biological side, it will be important to study the effectiveness of our method on other signaling pathways as well as metabolic and gene regulatory pathways.

ACKNOWLEDGEMENTS

We would like to thank Lisa Tucker-Kellogg for fruitful discussions, and the anonymous referees for their valuable comments. D. Hsu is supported in part by NUS grant R252-000-145-112. M.V. Clément is supported by grant R-185-000-106-213 from the National Medical Research Council (NMRC), Singapore.

REFERENCES

Alur,R., Belta,C., Ivančić,F., Kumar,V., Mintz,M., Pappas,G.J., Rubin,H. and Schug,J. (2001) Hybrid Modeling and Simulation of Biomolecular Networks. *Proc. 4th International Workshop on Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, Vol **2034**, 19–32.

Bellacosa,A., Chan,T.O., Ahmed,N.N., Datta,K., Malstrom,S., Stokoe,D., McCormick,F., Feng,J. and Tsichlis,P. (1998) Akt activation by growth factors is a multiple-step process: the role of the PH domain. *Oncogene*, **17**, 313–325.

Beyer,H.G. and Schwefel,H.P. (2002) Evolution strategies—A comprehensive introduction. *Natural Computing*, **1**, 3–52.

Chaudhary,A., King,W.G., Mattaliano,M.D., Frost,J.A., Diaz,B., Morrison,D.K., Cobb,M.H., Marshall,M.S. and Brugge,J.S. (2000) Phosphatidylinositol 3-kinase regulates Raf1 through Pak phosphorylation of serine 338. *Current Biology*, **10**, 551–554.

Chen,M. and Hofestaedt,R. (2003) Quantitative Petri net model of gene regulated metabolic networks in the cell. *In Silico Biology*, **3**, 347–365.

de Jong,H. (2002) Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *J. of Comp. Biol.*, **9(1)**, 67–103.

Ghosh,R. and Tomlin,C.J. (2001) Lateral Inhibition through Delta-Notch Signaling: A Piecewise Affine Hybrid Model. *Proc. 4th International Workshop on Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, **2034**, 232–246.

Hatakeyama,M., Kimura,S., Naka,T., Kawasaki,T., Yumoto,N., Ichikawa,M., Kim,J.H., Saito,K., Saeki,M., Shirouzu,M., Yokoyama,S. Konagaya,A. (2003) A computational model on the modulation of mitogen-activated protein kinase (MAPK) and Akt pathways in heregulin-induced ErbB signaling. *Biochem J.*, **373(2)**, 451–463.

Heldin,C.H. (2001) Signal Transduction: Multiple Pathways, Multiple Options for Therapy. *Stem Cells*, **19(4)**, 295–303.

Henzinger,T.A. (1996) The Theory of Hybrid Automata. *Proc. 11th Annual IEEE Symposium on Logic in Computer Science*, 278–292.

Horoszewicz,J.S., Leong,S.S., Kawinski,E., Karr,J.P., Rosenthal,H., Chu,T.M., Mirand,E.A. and Murphy,G.P. (1983) LNCaP model of human prostatic carcinoma. *Cancer Res.*, **43(4)**, 1809–1818.

Jiang,X. and Cheng,D.C. (2005) A Novel Parameter Decomposition Approach to Faithful Fitting of Quadric Surfaces. *Pattern Recognition: 27th DAGM Symposium, Lecture Notes in Computer Science*, **3663**, 168–175.

Khwaja,A. (1999) Akt is more than just a Bad kinase. *Nature*, **401**, 33–34.

Kimura,S., Hatakeyama,M., Kawasaki,T., Naka,T. and Konagaya,A. (2004) Parameter Estimation for the Simulation of Biochemical Pathways. *Proc. 15th IASTED International Conference on Modeling and Simulation*.

Kikuchi,S., Tominaga,D., Arita,M., Takahashi,K. and Tomita,M. (2003) Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics*, **19(5)**, 643–650.

Kirkpatrick,S., Gelatt Jr,C.D. and Vecchi,M.P. (1983) Optimization by Simulated Annealing. *Science*, **220(4598)**, 671–680.

Lincoln,P. and Tiwari,A. (2004) Symbolic Systems Biology: Hybrid Modeling and Analysis of Biological Networks. *Proc. 7th International Workshop on Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, **2993**, 660–672.

Matsuno,H., Tanaka,Y., Aoshima,H., Doi,A., Matsui,M. and Miyano,S. (2003) Biopathways representation and simulation on hybrid functional Petri net. *In Silico Biology*, **3**, 389–404.

Moelling,K., Schad,K., Bosse,M., Zimmermann,S. and Schweneker,M. (2002) Regulation of Raf-Akt Cross-talk. *The J. of Biol. Chem.*, **277(34)**, 31099–31106.

Moles,C.G., Mendes,P. and Banga,J.R. (2003) Parameter Estimation in Biochemical Pathways: A Comparison of Global Optimization Methods. *Genome Research*, **13(11)**, 2467–2474.

Nicholson,K.M. and Anderson,N.G. (2002) The protein kinase B/Akt signalling pathway in human malignancy. *Cellular Signalling*, **14**, 381–395.

Nagasaki,M., Doi,A., Matsuno,H. and Miyano,S. (2003) Genomic Object Net: I. A platform for modelling and simulating biopathways. *Applied Bioinformatics*, **2(3)**, 181–184.

Neapolitan,R.E. (2003) Learning Bayesian Networks. (1st Ed) *Prentice Hall*.

Pardalos,P.M. and Romeijn,H.E. (2002) Handbook of Global Optimization Volume 2. *Kluwer Academic Publisher*.

Reisig,W. (1992) A Primer in Petri Net Design. *Springer-Verlag*.

Sato,S., Fujita,N. and Tsuruo,T. (2004) Involvement of 3-Phosphoinositide-dependent Protein Kinase-1 in the MEK/MAPK Signal Transduction Pathway. *The J. of Biol. Chem.*, **279(32)**, 33759–33767.

Sorribas,A. and Savageau,M.A. (1988) Strategies for Representing Metabolic Pathways within Biochemical Systems Theory: Reversible Pathways. *Mathematical Biosciences*, **94(2)**, 239–269.

Teong,H.F.C. and Clément,M.V. (2006) Unpublished manuscript.

Tucker,W. and Moulten,V. (2005) Reconstructing Metabolic Networks Using Interval Analysis. *Lecture Notes in Computer Science*, **3692**, 192–203.

Vanhaesebroeck,B. and Alessi,D.R. (2000) The PI3K-PDK1 connection: more than just a road to PKB. *Biochem. J.*, **346**, 561–576.

Voss,K., Heiner,M. and Koch,I. (2003) Steady state analysis of metabolic pathways using Petri nets. *In Silico Biology*, **3**, 367–387.

Williams,B.C. and Millar,W. (1998) Decompositional, Model-based learning and its Analogy to Diagnosis. *AAAI/IAAI*, 197–204.

Ye,P., Entcheva,E., Grosu,R. and Smolka,S.A. (2005) Efficient Modeling of Excitable Cells Using Hybrid Automata. *Computational Methods in Systems Biology*.

Zevedei-Oancea,I. and Schuster,S. (2003) Topological analysis of metabolic networks based on Petri net theory. *In Silico Biology*, **3**, 323–345.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e281–e289
doi:10.1093/bioinformatics/btl247

Finding novel genes in bacterial communities isolated from the environment

Lutz Krause[1,*], Naryttza N. Diaz[1], Daniela Bartels[1], Robert A. Edwards[2,3,4], Alfred Pühler[6], Forest Rohwer[3,4], Folker Meyer[1] and Jens Stoye[5]

[1]Bielefeld University, Center for Biotechnology (CeBiTec) D-33594 Bielefeld, Germany, [2]Fellowship for Interpretation of Genomes, Burr Ridge IL, [3]Department of Biology, San Diego State University, San Diego, CA, [4]Center for Microbial Sciences, San Diego, CA, [5]Universität Bielefeld, Technische Fakultät D-33594 Bielefeld, Germany and [6]Universität Bielefeld, Lehrstuhl für Genetik, Fakultät für Biologie D-33594 Bielefeld, Germany

ABSTRACT

Motivation: Novel sequencing techniques can give access to organisms that are difficult to cultivate using conventional methods. When applied to environmental samples, the data generated has some drawbacks, e.g. short length of assembled contigs, in-frame stop codons and frame shifts. Unfortunately, current gene finders cannot circumvent these difficulties. At the same time, the automated prediction of genes is a prerequisite for the increasing amount of genomic sequences to ensure progress in metagenomics.

Results: We introduce a novel gene finding algorithm that incorporates features overcoming the short length of the assembled contigs from environmental data, in-frame stop codons as well as frame shifts contained in bacterial sequences. The results show that by searching for sequence similarities in an environmental sample our algorithm is capable of detecting a high fraction of its gene content, depending on the species composition and the overall size of the sample. The method is valuable for hunting novel unknown genes that may be specific for the habitat where the sample is taken. Finally, we show that our algorithm can even exploit the limited information contained in the short reads generated by 454 technology for the prediction of protein coding genes.

Availability: The program is freely available upon request.

Contact: Lutz.Krause@CeBiTec.Uni-Bielefeld.DE

1 INTRODUCTION

Novel sequencing methods have recently revolutionized the field of genome research. The sequencing of samples isolated directly from the environment allows access to organisms that can not be cultivated in the laboratory (Breitbart *et al*. (2002), Tyson *et al*. (2004), Venter *et al*. (2004)). Additionally, the massively parallel pyrosequencing system which was recently developed by 454 Life Science, Inc, has dramatically dropped the time and cost constraints of DNA sequencing (Margulies *et al*. (2005)). The application of 454 technology provides larger amounts of sequences at a lower cost compared to traditional DNA sequencing methods. These sequences are of great value for the identification of novel genes that can not be found in organisms cultured with traditional methods. The importance of such approaches is stressed by the fact that only a fraction of the living organism found in natural environments can be cultured by conventional methods (Tringe and Rubin (2005)).

The isolation and sequencing of DNA derived from diverse and mixed microbial communities is known as metagenomics, environmental genomics or ecogenomics. Although still in its infancy, this rapidly developing field has provided striking insights into the ecology and evolution of natural occurring microbial communities. Fields such as health and biotechnology have already benefited from metagenomics (Lombardot *et al*. (2006), Furrie (2006), Schloss and Handelsman (2003), Edwards and Rohwer *et al*. (2005), Edwards *et al*. (2006)).

Gene finding in environmental samples

Two different approaches are applied for predicting protein coding genes in bacterial genomes; intrinsic and extrinsic methods. Intrinsic methods (e.g. GLIMMER Delcher *et al*. (1999), GENEMARK Besemer and Borodovsky (1999)) analyze sequence properties of genomes to discriminate between coding sequences (CDS) and non-coding ORFs (NORFs). These methods exploit the different compositional properties of coding and non-coding sequences, which are mainly caused by a bias on codon usage in the CDS to optimize the translation efficiency (Gouy and Gautier (1982)).

In contrast, extrinsic methods (e.g. CRITICA Badger and Olsen (1999), ORPHEUS Frishman *et al*. (1998)) predict genes by searching for stretches of DNA that were conserved during evolution. The success of extrinsic methods can be explained by the fact that during evolution most of the new genes are formed by duplication, rearrangement and mutation events of existing genes (Chothia *et al*. (2003)).

The prediction of protein coding genes in environmental samples is problematic for several reasons. One is the low sequence quality of the assembled contigs which may lead to frame shifts and in-frame stop codons in the CDSs contained therein. Another problem is that assembled contigs may be too short to reveal the genome specific sequence properties, which are crucial in the application of intrinsic gene prediction methods. These reasons limit their application to environmental samples. Currently, the majority of the CDSs in environmental samples are identified based on a BLAST search against databases of known proteins.

*To whom correspondence should be addressed.

Species that are abundant in natural environments will also be over-represented in the samples. These species do not represent a problem while assembling them, and large stretches of their genomes can be obtained. But, the under-represented species constitute a challenge since for those only short contigs with low coverage are obtained. One problem related to the low coverage is that these contigs are even more prone to contain in-frame stop codons or frame shifts. Therefore, applying existing gene finders to environmental samples is fraught with difficulties because they were not designed to cope with this type of errors and short contigs.

Strategy

The main idea for the novel gene prediction method presented in this work is to search for stretches of DNA that are conserved within the environmental sample. Here, the algorithm does not rely on a pairwise sequence comparison, but instead it combines information from all BLAST hits at the same time. Conserved coding sequences are discriminated from conserved non-coding regions based on their synonymous substitution rate.

In functional proteins, the coding genes show a much higher number of synonymous substitutions than in non-coding sequences. The rate of synonymous to non-synonymous substitutions (k_S/k_A) reflects the interchange of positive selection and neutral evolution. Therefore, investigating the number of synonymous and non-synonymous substitutions can supply valuable information on whether or not a sequence stretch is under constraint for functional selection. This information can be used for the identification of genes in bacterial and eukaryotic genomes (Badger and Olsen (1999), Nekrutenko *et al*. (2003a), Nekrutenko *et al*. (2003b) and Moore and Lake (2003)).

For the prediction of protein coding sequences contained in a contig from an environmental sample, first a BLAST search against a nucleotide database is conducted. For this in principle any nucleotide database can be used, e.g databases containing complete genomes, metagenomes or known genes. To search for novel habitat specific genes a BLAST search against a database that exclusively contains all sequences from that sample can be employed. Subsequently, the algorithm needs to discriminate if the BLAST hits match conserved coding sequences, conserved non-coding regions, or shadows of CDSs in another reading frame. Additionally, a CDS may be embedded in long BLAST hits. For this case, the gene boundaries need to be identified. Given all BLAST hits for a contig, the algorithm will find the best path through all hits at the same time. In order to accomplish this task, several different features are taken into account: (a) the synonymous substitution rate at each position in the contig, (b) the positions of stop codons in the contig and (c) the position of stop codons in matching database sequences. Additionally, the end of BLAST hits are considered as possible indications for the boundaries of coding regions.

In the gene prediction process the algorithm will avoid in-frame stop codons, but otherwise will favor regions with a high synonymous substitution rate. The outcome of the BLAST hits are used to assign six scores to each nucleotide, one for each of the six possible reading frames, reflecting the nucleotides coding potential in this reading frame. Scores are assigned by counting the number of synonymous and non-synonymous substitutions at each position for each of the six reading frames. As a result, a scoring matrix with scores for each nucleotide in the contig is obtained. Based on these scores, a dynamic programming method is applied to find the optimal path through the matrix that maximizes the overall score (the sum of all scores on the path). The usage of a combined score for all BLAST hits should result in a superior performance compared to methods that rely on simple pairwise sequence alignments. The advantage should be particularly profound when a database of low quality with short contigs and many frame shifts is used for the BLAST based search for conserved sequences.

2 METHODS

The gene prediction algorithm

The algorithm can be divided into four phases: (1) a BLAST based search for conserved sequences (2) the calculation of combined scores (3) the prediction of coding sequences by dynamic programming and (4) the postprocessing.

Phase 1: Blast based search for conserved sequences During the first phase of the algorithm a BLAST search against a nucleotide database is conducted. Hereby, the contig as well as all sequences in the database are translated into all six reading frames (if the database contains known genes only the contig will be translated into all six reading frames). As the BLAST search is conducted on the amino acid level, each obtained hit is associated with a specific reading frame in the contig. The BLAST hits obtained are filtered, hits with $k_S/k_A < 1$ are excluded from the subsequent analysis as these do not indicate the presence of a coding sequence.

Phase 2: Calculation of combined scores In the second phase of the algorithm, the remaining hits are used to assess the coding potential of each nucleotide in the contig. Given a contig c of length n, $c[i]$ denotes the nucleotide at position i of that contig ($1 \leq i \leq n$). A nucleotide $c[i]$ could be coding in one of the six reading frames $k \in \{-3, -2, -1, +1, +2, +3\}$, or non-coding, denoted by $k = 0$. For each position i and for each reading frame k, the number of synonymous and non-synonymous substitutions at position i are counted (Figure 1). This is done by comparing the nucleotide sequence of the contig to the nucleotide sequence of all BLAST hits in this reading frame. The number of synonymous and non-synonymous substitutions are used to score that $c[i]$ is coding in reading frame k. Synonymous substitutions contribute with a positive score, non-synonymous substitutions with a negative score. Additionally, the correct ends of the coding sequences need to be determined. Therefore, stop codons in the contig are penalized with a negative score in the according frame. For a given BLAST hit both the contig and the matching database sequence of the BLAST hit may contain stop codons. To discriminate between real stop codons and stop codons introduced by sequencing errors, additionally negative scores are applied for: (a) all stop codons in the database sequences of the BLAST hits, (b) for ends of BLAST hits, as these also may indicate the boundaries of genes (Figure 1). Subsequently, each score obtained is normalized by the number of hits that contribute to that score. Using this strategy for all BLAST hits in reading frame k, a single combined score that reflects the coding potential of the contig at position i in this reading frame is derived. Additionally, for $k = 0$ a score of zero is assigned to each position i of the contig. As a result, a scoring matrix s_{ik} is derived which provides a position specific score that the contig is coding in one of the six reading frames or non-coding (Figure 1).

Phase 3: Prediction of coding sequences Coding sequences are predicted in the third phase. To assign one of the six reading frames k (or $k = 0$ for non-coding) to each position of the contig, the algorithm searches for the path in the scoring matrix s_{ik} that maximizes the sum of all scores on the path. According to the optimal path, each position i of the contig is subsequently labeled with the frame k it passes through at this position. Depending on their reading frame, genes may only start or

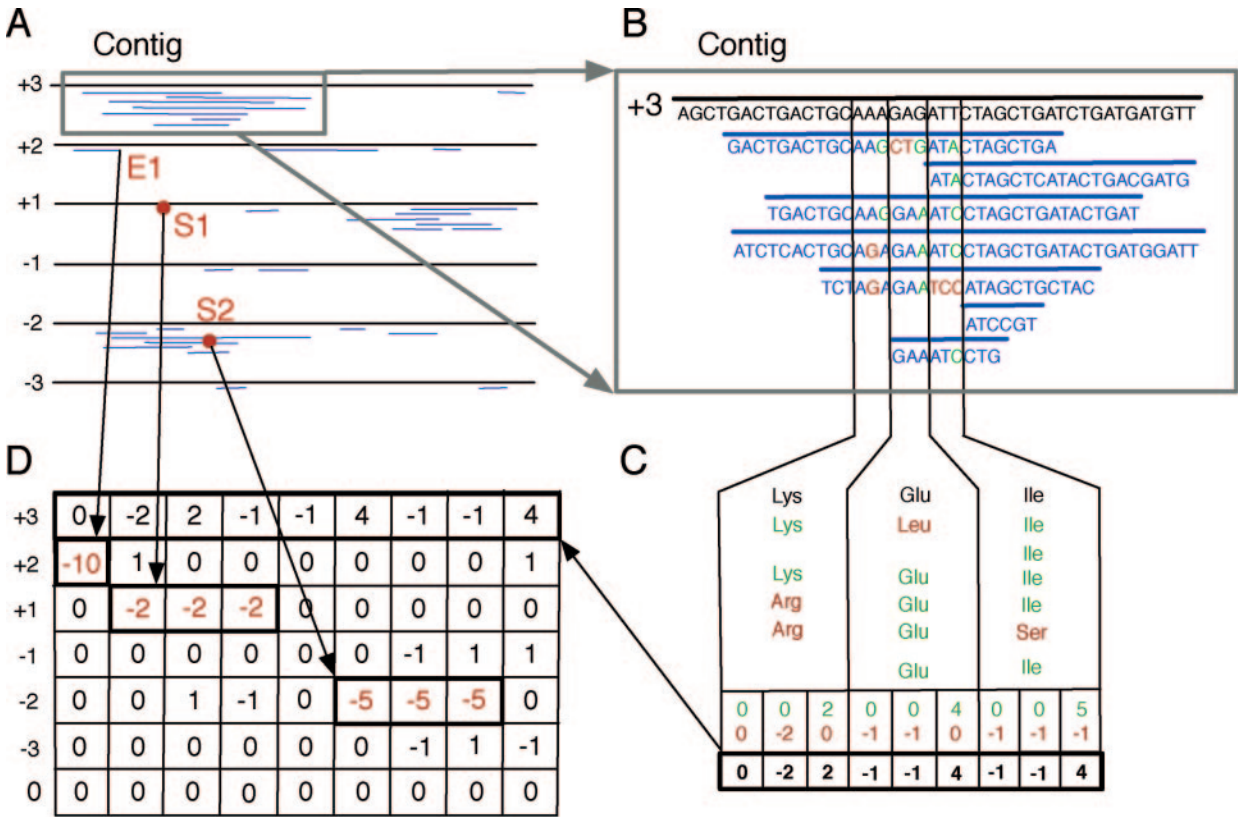


Fig. 1. Calculating combined scores. All scores are depicted without normalization. **A**) all six reading frames of a contig are shown (the continuous lines). BLAST hits matching the respective reading frames are displayed as blue bars below the reading frame. **B**) The nucleotide sequence of each reading frame of the contig is compared with all database sequences matching this reading frame. The number of synonymous and non-synonymous substitutions at each position is used as a score that the contig at this position is coding in the respective reading frame. **C**) The number of synonymous substitutions at each position are used as a positive score. The number of non-synonymous substitutions at each position contribute with a negative score. **D**) The calculated scores for each position and reading frame are stored in a matrix. For $k = 0$ a score of zero is assigned to each position i of the contig. Penalties are additionally added to the respective position and reading frame for stop codons in the contig (S1), in the matching database sequence (S2) as well as for the end of BLAST hits (E1).

stop at certain positions. Therefore, a valid path may not jump arbitrarily between frames, but instead underlies certain restrictions. To be precise, the set $V(i, k)$ of all valid precursors of a frame k at position i is defined as:

$$V(i,k) = \begin{cases} \{j, 0, -j\} & \text{if} \quad k = 0 \\ \{k, 0, -k\} & \text{if} \quad |k| = j \\ \{k\} & \text{otherwise.} \end{cases}$$

where $j = (i - 1) \bmod 3 + 1$. Figure 2 depicts the scoring matrix of combined scores and the calculation of the optimal path. This figure also introduces several terms used in the following. The optimal valid path for a scoring matrix s_{ik} can be calculated using dynamic programming by the following recursion:

$$f_i(k) = \max_{k' \in V(i,k)} \begin{cases} f_{i-1}(k') + s_{ik} + 2q & \text{if } k < 0 \text{ and } k' > 0 \\ f_{i-1}(k') + s_{ik} + q & \text{if } k \neq 0 \text{ and } k' \neq k \\ f_{i-1}(k') + s_{ik} & \text{otherwise} \end{cases}$$

where q is a negative score that is added to leave a gene on the forward strand or to enter a gene on the reverse strand ($2q$ are added if a gene on the forward strand is left and a gene on the reverse strand is entered at the same time). Thus, q is added for each 5′ end of a gene on a path. The penalty q was introduced to predict genes only in areas with sufficient coding evidence. The calculated value $f_i(k)$ is the maximal score of all paths that enter s at position 1 and pass through k at position i.

Phase 4: Postprocessing During the post-processing phase, the predictions are joined and frame shifts are identified. When a BLAST search against a database of short contigs is employed, genes may be covered only partially by hits which may result in the prediction of several fragments. This is particularly profound when a BLAST search against reads provided by the 454 technology is conducted. Therefore adjacent predictions within the same reading frame are joined if (a) their distance on the contig does not exceed 400 bp and (b) the sequence of the contig that separates the predictions does not contain an in-frame stop codon.

To identify frame shifts that were introduced by sequencing errors all adjacent predictions located on the same strand but within a different reading frame are predicted as frame shifts if (a) their distance on the contig is less than 200 bp and (b) they do do not have an in-frame stop codon close to the potential frame shift. As an optional postprocessing step, our algorithm can also extend predicted CDS to the longest possible ORF available for that prediction.

Implementation

The algorithm was implemented in PERL using an object oriented approach.

Measuring the performance

To evaluate the performance of the novel gene finder predictions were compared to known annotated genes. For this purpose, two measurements

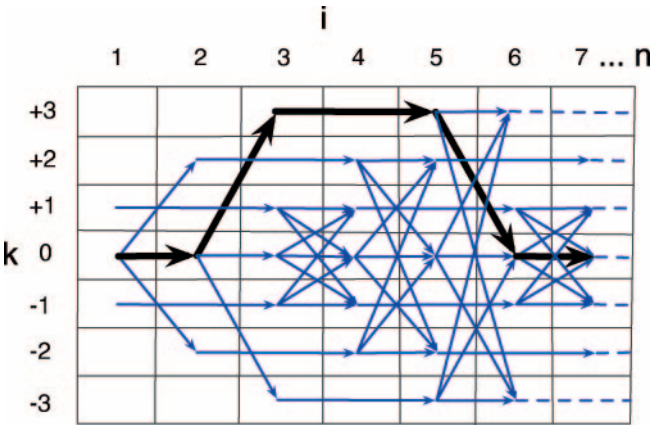


Fig. 2. Predicting coding sequences by calculating the optimal path in scoring matrix of combined scores. This figure shows the scoring matrix s_{ik} for the first seven positions of a contig. All valid paths in the scoring matrix are indicated with arrows. A gene is entered, if a path passes through frame $k \neq 0$ with the precursor frame $k' \neq k$. Accordingly, a gene is left, if a path that comes from a precursor frame $k' \neq 0$ enters a frame $k \neq k'$. The bold arrows depict an example path predicting a 3 bp long gene on the reading frame +3

are widely used: sensitivity and specificity. Sensitivity is a measure of the ability of the algorithm to predict known genes and is defined by $Sens = \frac{TP}{TP+FN}$. The specificity is a measure of the reliability of the predictions, given by the ratio $Spec = \frac{TP}{TP+FP}$. For the evaluation of the performance predictions were extended to the next 5′ stop codon. If an annotated CDS ends at that stop codon the prediction was counted as true positive (TP). Otherwise, the prediction was regarded as a false positive. All genes that are not completely embedded in the contigs are named truncated genes. These genes may appear at the end or beginning of the assembled contigs, therefore lacking the start or termination site of the gene. Truncated genes were excluded from the analysis.

Training GLIMMER on a synthetic metagenome

The prokaryotic gene finder GLIMMER version 3.01b was used to predict the genes of a synthetic metagenome (described in Materials). For the training step, all fragments of this metagenome were chained to one continuous contig. Adjacent fragments were concatenated with a linker sequence containing a stop codon in each of the six reading frames. Subsequently the GLIMMER ICM model was trained on the chained contig.

3 MATERIALS

Metagenome obtained with pyrosequencing

The performance of the algorithm was evaluated on a metagenome of a bacterial community isolated from the Solar Salterns in San Diego, CA (B. Rodriguez-Brito, R. Edwards, and F. Rohwer, Unpublished). Total community DNA was purified as described elsewhere (Edwards *et al*. (2006)) and sequenced using pyrosequencing by 454 Life Sciences, Inc, (Branford, CT). Using the 454 technology ≈60 Mb were obtained with an average read length of 100 bp. The reads were assembled using Phrap (Green (1994)). This resulted in 80,878 contigs with 16 Mb in total. In the following, this set is called *all contigs*. From this set a subset of contigs longer than 1,000 bp (2,244 contigs with 3.8 Mb in total) was selected, called hereafter *long contigs*.

Table 1. Annotated and published genomes used to create a synthetic metagenome

Organism	Accession number
Bacteria	
Alphaproteobacteria	
Candidatus pelagibacter ubique HTCC1062	NC_007205
Rhodobacter sphaeroides 2.4.1 chromosome 1	NC_007493
Gammaproteobacteria	
Shewanella oneidensis MR-1	NC_004347
Thiomicrospira crunogena XCL-2	NC_007520
Vibrio cholerae O1 biovar eltor str. N16961 chromosome 1	NC_002505
Cyanobacteria	
Prochlorococcus marinus subsp. *pastoris* str. CCMP1986	NC_005072
Synechococcus sp. WH 8120	NC_005070
Archaea	
Euryarchaeota	
Pyrococcus horikoshii OT3	NC_000961
Crenarchaeota	
Sulfolobus solfataricus P2	NC_002754

Species names and accessions numbers downloaded form the NCBI database.

The environmental sample from the Sargasso Sea

For the prediction of protein coding genes in metagenomes, the environmental sample from the Sargasso Sea (Venter *et al*. (2004)) was used as BLAST database during the search for conserved regions in the first phase of the algorithm. To save computational time, only half (≈390 Mb) of the entire Sargasso Sea sample was used.

Generating a synthetic metagenome

As a proof of concept, the algorithm was evaluated on a set of nine completely sequenced and annotated genomes (seven Bacteria and two Archaea, see Table 1). Members from the alpha, gammaproteobacteria and cyanobacteria groups were selected as they were reported to be abundant in the Sargasso Sea sample (Venter *et al*. (2004)). We also added Archaea to the evaluation set because they can be regarded as under-represented species in surface water marine environments. All genomic sequences and their respective annotations were downloaded from the NCBI Reference Sequence database (RefSeq) release 15 (Pruitt *et al*. (2005)). A synthetic metagenome with known CDSs was created by splitting the genome of each of the nine organism into fragments of length 4000 bp. A subset of *non-hypothetical* genes was created based on the annotated gene products from the public annotations. In this set all annotated genes with a gene product description of 'hypothetical protein' were excluded. Additionally, artificial sequencing errors (frame shifts and in-frame stop codons) were incorporated into all genes of the synthetic metagenome. In order to perform a systematic evaluation, all artificial sequencing errors were added to the synthetic metagenome in a controlled way. In one experiment, in-frame stop codons were added to the center of each gene of the original synthetic metagenome. In a second experiment frame shifts were incorporated to the center of all genes of the original synthetic metagenome.

4 RESULTS

Gene prediction in a synthetic metagenome using the environmental sample from the Sargasso Sea

Our algorithm can be used to identify genes contained in an environmental sample by directly searching for conserved regions within the sample. This approach may elucidate novel unknown genes present in the environmental sample which may be specific for the habitat the sample was taken from. The performance of the algorithm of predicting genes in an environmental sample by running a BLAST search against the sample itself was evaluated on the environmental sample data from the Sargasso Sea (Venter *et al*. (2004)). But, instead of drawing the contigs for which the genes are predicted directly from the Sargasso Sea sample, we used a more controlled and reliable data set. We chose several completely sequenced and annotated genomes from Bacteria groups that were also reported to be present in the species composition of the Sargasso Sea sample (Venter *et al*. (2004)). The genomes of these organisms were split into fragments of size 4000 bp, together forming a synthetic metagenome as a reliable standard of truth. Subsequently the genes of these contigs were predicted with our algorithm based on a BLAST search against the Sargasso Sea sample. To accurately evaluate the prediction performance that can be expected for a 'real' metagenome, sequences from alpha and gammaproteobacteria which are reported as over-represented in the Sargasso Sea sample, cyanobacteria which are modest abundant, as well as sequences from extremely scarce groups (two Archaea members) were included. The performance of the algorithm was measured by comparing the genes predicted for the synthetic metagenome with the known genes from the public genome annotations. To additionally evaluate the performance for sequencing errors that may frequently occur in metagenomes, three validation sets were used: (1) synthetic metagenome without artificial sequencing errors, (2) synthetic metagenome with in-frame stop codons and (3) synthetic metagenome with frame shifts.

Experiment 1: Gene prediction in a synthetic metagenome without artificial sequence errors The sensitivity and specificity reached by the algorithm for each organism contained in the synthetic metagenome is shown in Table 2. The results show that the sensitivity of the method strongly depends on the abundance of the different groups of Bacteria in the sample. While for the more abundant alpha, gammaproteobacteria and cyanobacteria an average sensitivity of 79% for all genes and 89% for the subset of non-hypothetical genes is achieved, for the Archaea the sensitivity is strongly reduced. The lower sensitivity for the Archaea was expected because this group is very rare in surface water marine environments and hence extremely scarce in the environmental sample from the Sargasso Sea. For the two cyanobacteria contained in the synthetic metagenome even a sensitivity of more than 94% is achieved for the non-hypothetical genes. In contrast, with a specificity between 88% and 99% the algorithm is highly specific for all groups. On average the specificity is 95%. The considerably lower overall sensitivity ($Sens_{all}$) when compared to the sensitivity for the subset of non-hypothetical genes ($Sens_{nh}$) can be explained by the fact that most of the genes labeled as 'hypothetical protein' in the public annotations were originally predicted with intrinsic methods. Many of these genes are either orphans (genes without sequence similarity to any known gene) or in fact non-coding and hence wrong annotations.

Table 2. Performance for a synthetic metagenome evaluated on the Sargasso Sea environmental sample

Organism	$Sens_{all}$	$Sens_{nh}$	Specificity
Bacteria			
Alphaproteobacteria			
C. pelagibacter	91.07	93.76	97.63
R. sphaeroides	62.01	77.62	97.02
Gammaproteobacteria			
S. oneidensis	85.36	95.12	90.44
T. crunogena	65.38	79.33	97.54
V. cholerae	69.66	87.66	93.88
Cyanobacteria			
P. marinus	93.29	94.42	89.75
Synechococcus sp.	82.99	94.13	87.83
Archaea			
Euryarchaeota			
P. horikoshii	29.99	66.40	97.77
Crenarchaeota			
S. solfataricus	26.69	43.44	98.89
Average	67.38	81.32	94.53

$Sens_{all}$ refers to the sensitivity calculated over all genes contained in the synthetic metagenome. $Sens_{nh}$ is the sensitivity calculated over all non-hypothetical genes. The entire Bacteria group represents the most common organisms in the Sargasso Sea sample. While the Archaea is the extremely scarce set for surface water marine environment.

Experiment 2: Gene prediction in a synthetic metagenome with artificial in-frame stop codons In the second experiment the performance of the algorithm was evaluated on genes containing in-frame stop codons. Therefore, an in-frame stop codon was added to the center of each annotated gene in the synthetic metagenome. In addition to the sensitivity and specificity, the percentage of true positives (TP) that span the artificially added stop codons was measured. In comparison to the synthetic metagenome without artificial sequence errors, for the genes with in-frame stop codons only a slight reduction in sensitivity and specificity was registered. The sensitivity is reduced by 1.7% for all genes and 1.3% for the subset of non-hypothetical genes. The reduction in specificity is 0.3%. On average, for 77% of all identified genes (TP) the prediction also spans the added in-frame stop codon (Table 3) and therefore correctly recognizes the stop codon as sequencing error. Strikingly, for the synthetic metagenome without artificial sequence errors only 4 predictions wrongly span a 'real' stop codon terminating the translation. These results demonstrate that the algorithm is quite robust for the task of identifying functional genes containing in-frame stop codons, generated by sequencing errors. These results also reveal the strength of our method to incorporate several features to determine the boundaries of coding sequence and to discriminate between 'real' stop codons and those introduced by sequencing errors.

Experiment 3: Gene prediction in a synthetic metagenome with artificial frame shifts In the third experiment the performance of the novel algorithm to predict frame shifts introduced by sequencing errors was evaluated. Therefore, an artificial frame shift was

Table 3. Performance for a synthetic metagenome with artificial in-frame stop codons

Organism	$Sens_{all}$	$Sens_{nh}$	Spec	SC predicted
Bacteria				
Alphaproteobacteria				
C. pelagibacter	88.87	92.20	97.58	71.71
R. sphaeroides	61.62	77.18	97.33	73.10
Gammaproteobacteria				
S. oneidensis	83.90	94.24	89.58	82.15
T. crunogena	64.95	79.01	97.88	80.23
V. cholerae	68.89	86.94	94.00	79.52
Cyanobacteria				
P. marinus	88.97	90.18	88.51	74.71
Synechococcus sp.	78.74	92.69	85.65	70.75
Archaea				
Euryarchaeota				
P. horikoshii	29.27	65.20	98.69	80.97
Crenarchaeota				
S. solfataricus	25.96	42.71	99.02	81.41
Average	65.69	80.04	94.25	77.17

$Sens_{all}$ is the sensitivity calculated over all genes contained in the synthetic metagenome. $Sens_{nh}$ is the sensitivity calculated over the subset of all non-hypothetical genes. SC predicted: percentage of true positives (TP) that correctly span in-frame stop codons.

Table 4. Performance for a synthetic metagenome with artificial frame shifts

Organism	$Sens_{all}$	$Sens_{nh}$	Spec	Percentage of TP predictions correctly identified as frame shift
Bacteria				
Alphaproteobacteria				
C. pelagibacter	86.39	89.32	97.73	57.71
R. sphaeroides	56.80	72.08	98.03	92.22
Gammaproteobacteria				
S. oneidensis	81.15	90.93	93.02	68.65
T. crunogena	59.69	73.33	98.33	66.67
V. cholerae	71.54	83.05	96.19	69.11
Cyanobacteria				
P. marinus	84.65	88.88	92.62	58.77
Synechococcus sp.	72.46	90.39	91.02	79.19
Archaea				
Euryarchaeota				
P. horikoshii	26.09	54.42	96.45	54.64
Crenarchaeota				
S. solfataricus	23.23	39.41	98.80	48.51
Average	62.44	75.76	95.80	66.16

$Sens_{all}$ is the sensitivity calculated over all genes contained in the contigs. $Sens_{n\ h}$ is the sensitivity calculated over the subset of non-hypothetical genes

added to each of the genes of the synthetic metagenome. For this data set, those predictions that do not match a fragment of an annotated gene where counted as false positives (FP). For those annotated genes of which at least one of its fragments is identified were counted as true positives (TP). Compared to the synthetic metagenome with no artificial mutations, the sensitivity and specificity is again only slightly reduced (Table 4). For this data set, 66% of the identified genes (TP) were also correctly predicted to have a frame shift. Noteworthy, for the synthetic metagenome without artificial errors only 357 frame shifts out of 11,686 true positive predictions were registered. This finding shows the high reliability of the method to predict frame shifts. As for the above experiments, the specificity values obtained by each genome are high, the average specificity value is 95%.

Gene identification in environmental samples obtained by 454 technology

At present, the main drawback of the recently developed high throughput parallel pyrosequencing is the short length of the reads obtained (≈100 bp on average). This is particularly undesirable when dealing with environmental data sets, since the sample is a large mixture of different species. To verify whether our algorithm is still able to identify genes in metagenomes obtained with the 454 technology, we assembled the 454 reads from the Solar Salterns sample into contigs and predicted the genes for the subset of all *long contigs*. For this verification we performed two experiments: First, a BLAST search against a database made from the set of *all contigs* from the Solar Salterns sample was conducted. Second, a direct BLAST search against a database of all 454 reads without prior assembly was employed. To validate the outcome from both experiments the respective predictions (extended to the longest possible ORF for that prediction) were compared with known proteins from the KEGG database (Ogata *et al*. (1999)) using BLAST.

Table 5. KEGG supported predictions. Number of predicted genes for a metagenome sequenced with 454 technology that have hit in the KEGG database.

Database	Number of predictions	Number of predictions with E-value up to 10^{-50}	10^{-20}	10^{-10}	10^{-5}
KEGG					
Contigs	3219	467	1544	2451	2858
Reads	3496	556	1699	2585	3044

Assembled contigs and 454 reads without prior assembly were used for BLAST search.

For both experiments, a high fraction of the predicted genes has significant BLAST hits against known proteins from the KEGG database. Remarkably, the number of predicted genes for the reads without assembly does not differ much when compared to the contigs (see Table 5). It should be pointed out that when looking at the BLAST hits against the KEGG database it seems that many of the predicted genes are fragmented due to internal frame shifts. Therefore during the BLAST search against the KEGG database, weaker E-values are obtained for these fragments. The predicted genes that do not match any known protein in the KEGG database constitute an interesting set for further studies as they could be either of false predictions, known genes with no or only a weak sequence similarity to the genes contained in KEGG, or more interestingly novel unknown genes. These results for the Solar Salterns sample demonstrate that the novel algorithm is well

Table 6. Performance for a synthetic metagenome evaluated on sequences obtained by pyrosequencing

Organism	$Sens_r$	$Sens_c$	$Sens_{nhr}$	$Sens_{nhc}$	$Spec_r$	$Spec_c$
Bacteria						
Alphaproteobacteria						
C. pelagibacter	38.96	28.02	43.92	31.66	85.29	91.54
R. sphaeroides	27.74	19.13	38.03	26.81	70.67	87.18
Gammaproteobacteria						
S. oneidensis	25.37	16.19	38.23	25.12	80.86	88.21
T. crunogena	36.21	23.49	46.31	30.29	86.42	93.43
V. cholerae	29.71	18.84	43.19	28.69	82.22	90.70
Cyanobacteria						
P. marinus	30.40	20.94	43.08	29.91	89.67	91.53
Synechococcus sp.	23.24	17.01	43.67	31.82	77.14	87.73
Archaea						
Euryarchaeota						
P. horikoshii	33.61	31.87	66.80	62.60	89.64	94.67
Crenarchaeota						
S. solfataricus	26.35	25.15	41.97	41.32	90.34	95.46
Average	30.18	22.29	45.02	34.25	83.58	91.16

$Sens_r$ and $Sens_c$ is the sensitivity for the synthetic metagenome when blasting against all 454 reads or against all assembled contigs. $Sens_{nhr}$ and $Sens_{nhc}$ is the sensitivity calculated for the subset of non-hypothetical genes of the synthetic metagenome when a BLAST search is done against the 454 reads and assembled contigs, respectively.

suited to predict genes in 'real' metagenomes, even if these samples are sequenced using the 454 technology.

Gene prediction in synthetic metagenomes using contigs and reads derived by pyrosequencing

We further evaluated the performance of the new gene finding algorithm for sequences obtained with the 454 technology (see Table 6), taking the synthetic metagenome dataset as a controlled standard of truth. The genes were predicted for the synthetic metagenomes dataset by employing a BLAST search against two different databases: one containing all assembled contigs from the Solar Salterns sample, and another containing all unassembled reads from the same sample.

In respect to the small size of the database used in the BLAST search (16 Mb for the assembled contigs and 60 Mb for the reads without prior assembly) the sensitivity obtained is very good. The highest sensitivity is reached for *Pyrococcus horikoshii*, 67% and 63% (for the subset of all non-hypothetical genes) calculated for the reads without assembly and the assembled contigs, respectively. Interestingly, these findings indicate that in contrast to the sample from the Sargasso Sea, the Archaea group is more abundant in the sample from the Solar Salterns. A second interesting observation is the good performance when running BLAST against the 454 reads without assembly, despite the fact that the average length of the reads is 100 bp. A specificity of 84% is achieved on average. Moreover, when compared to the assembled contigs the sensitivity is increased by ≈11%. In particular, these results for the short 454 reads reveal one of the strengths of our method: to consider all BLAST hits at the same time by calculating the optimal path through the matrix of combined scores instead of analyzing simple pairwise BLAST hits. This strategy allows us to identify genes that get only several short hits, even if all of the single hits are not significant.

Yet, determining the correct boundaries of the CDS when running BLAST against a small database of 454 reads is difficult, many genes are only partially covered by hits. As an optional postprocessing step our algorithm therefore can automatically extend predictions to the longest possible ORF.

Time efficiency of the novel algorithm

The running time of the novel algorithm highly depends on the size of the BLAST database since most of the running time is consumed during the BLAST based search for conserved regions, for the parsing of BLAST results as well as for the calculation of combined scores. For the evaluation presented in this survey all runs of the algorithm were executed on a compute cluster located at the Center of Biotechnology (CeBiTec), Bielefeld University. The cluster is composed of 128 Sun Fire V20z nodes. Each node has two 1.8 GHz AMD Opteron 244 CPUs and 2 Gb of RAM. The overall running time was 1 hour and 50 minutes for predicting the genes of the synthetic metagenome (≈24 Mb) when a BLAST search against half of the Sargasso Sea sample (≈390 Mb) was employed. The running time in average is 28s for the BLAST search, 17s for parsing the BLAST results and calculating the combined scores and 1s for predicting coding sequence by dynamic programming and postprocessing for a 4 Kb fragment when run on a single node using one CPU.

GLIMMER performance on synthetic metagenome

Most of the contemporary gene finding methods model frequencies of short oligonucleotides to discriminate between coding and non-coding sequences (e.g. by using a Markov chain or a Hidden Markov model). Before these methods can be used for gene prediction, usually as a first step the model needs to be trained to learn the organism specific sequence composition of the genome under study. As most of these methods model average sequence properties they may fail to adequately learn the oligonucleotide frequencies of diverse microbial assemblages. Pitfalls of existing gene finding technologies were examined by employing the state-of-the art microbial gene finder GLIMMER as an example. GLIMMER was trained on the synthetic metagenome itself as described in the Methods section. Subsequently, the trained GLIMMER was applied on each fragment. Although GLIMMER is very accurate for complete genomes (http://www.cbcb.umd.edu/software/glimmer/) the accuracy for the synthetic metagenome is strongly reduced (Table 7). Table 7 also points to one substantial problem that may affect intrinsic methods when applied to environmental data: the diverse compositional biases of different organisms contained in the sample. Another problem may be the unequal abundance of species, as overrepresented species have a stronger influence during training which may result in an unbalanced model. Also the synthetic metagenome on which GLIMMER was trained is unbalanced as it contains fragments from seven genomes with a low GC content and from two genomes with a high GC content (GC > 55%). The average GC content is ≈47%. The prediction accuracy of GLIMMER for the synthetic metagenome strongly depends on whether the fragments come from a genome with a high or low GC content. While GLIMMER has a good performance for the genomes with a low GC content, for the two genomes with a high GC content the performance is highly reduced. For the

Table 7. GLIMMER performance for a synthetic metagenome

Organism	Contig size (Mb)	GC (%)	$Sens_{all}$	$Sens_{nh}$	Spec
C. pelagibacter	1.3	30	94.34	95.54	78.39
P. marinus	1.7	31	91.80	94.87	74.04
S. solfataricus	3.0	36	91.46	93.89	72.29
P. horikoshii	1.7	42	88.28	95.60	76.92
T. crunogena	2.4	43	98.72	99.12	71.10
S. oneidensis	5.0	46	95.61	98.08	67.22
V. cholerae	3.0	48	90.68	98.48	69.52
Synechococcus sp.	2.4	59	43.80	51.60	60.41
R. sphaeroides	3.2	69	12.16	14.65	23.69
Average	2.6	47	78.53	82.42	65.95

Genomes ordered by GC content. GLIMMER was trained on the synthetic metagenome itself. $Sens_{all}$ and $Sens_{n\ h}$ is the GLIMMER sensitivity for set of all and for the subset of non-hypothetical genes. Spec: Specificity

genome with the highest GC content (*R. sphaeroides*) the accuracy is close to the one expected by a random decision drawn by a flipping a coin experiment. Owing to the diverse composition, high species richness and unequal species abundance, real metagenomes isolated from natural occurring organism assemblages possess a considerably higher complexity than the synthetic metagenome used in this study. Therefore, it is reasonable to expect that for real metagenomes the problems that affect intrinsic methods should be even more profound.

5 DISCUSSION

In this paper we presented a novel algorithm that was designed to predict genes in environmental samples. The algorithm is robust for the most common problems encountered when predicting genes in these data sets: short length of the assembled contigs and a low sequence quality.

Although, the focus of the algorithm is directed on the detection of novel genes, our algorithm can also be used to identify known genes in environmental samples: instead of searching against a database containing all fragments from the environmental sample a direct search against a database containing the sequences of known genes can be conducted.

Our results show that for large samples like the Sargasso Sea, a high fraction of the gene content can be identified based on the search for sequence conservation within the sample.

The results further demonstrate that even the short reads obtained by pyrosequencing can be used to identify protein coding genes. Therefore, environmental samples sequenced with the 454 technology may be a valuable resource to identify unknown (habitat-specific) genes. To search for novel genes our algorithm requires that at least a fraction of reads is assembled into contigs. Subsequently the complete database of reads can be used to predict the genes of these contigs. Our results therefore suggest the following strategy to identify novel (habitat-specific) genes in environmental samples: to sequence part of the sample with conventional methods to obtain longer fragments that can be assembled into contigs and additionally to sequence large amounts of data at low cost with the 454 technology to increase the size of the database that can be used to search for conserved sequences.

As our method relies on sequence similarities for the prediction of protein coding genes when running BLAST against the sample itself, the method strongly depends on the size and species composition of the sample. The sensitivity of the algorithm may be improved by incorporating general sequence properties of coding sequences or proteins.

ACKNOWLEDGEMENTS

LK was supported by the DFG Graduiertenkolleg 635 Bioinformatik. RAE and FR were supported by a grant NSF DEB-BE 04-21955 from the NSF Biocomplexity program. We thank Beltran Rodriguez-Brito for generating the environmental data. NND was supported by the Deutscher Akademischer Austausch Dienst. Thanks to the anonymous reviewers for valuable comments and helpful remarks.

REFERENCES

Badger,H. and Olsen,G.J. (1999) Article title. *Mol. Biol. Evol*, **16**, 512–524.

Besemer,J. and Borodovsky,M. (1999) Heuristic approach to deriving models for gene finding. *Nucleic Acids Res.*, **27**, 3911–3920.

Ewing,B., Hillier,L., Wendl,M.C. and Green,P. (1998) Base-calling of automated sequencer traces using phred. I. *Genome Res.*, **8**, 175–185.

Breitbart,M., Salamon,P., Andresen,B., Mahaffy,J.M., Segall,A.M., Mead,D., Azam,F. and Rohwer,F. (2002) Genomic analysis of uncultured marine viral communities. *Proc Natl Acad Sci U S A*, **99**, 14250–14255.

Chothia,C., Gough,J., Vogel,C. and Teichmann,S.A. (2003) Evolution of the protein repertoire. *Science*, **300**, 1701–1703.

Delcher,A.L., Harmon,D. and Kasif,S. (1999) Improved microbial gene identification with GLIMMER. *Nucleic Acids Res.*, **27**, 4636–4641.

Edwards,R.A. and Rohwer,F. (2005) Viral metagenomics. *Nat Rev Microbiol*, **3**, 504–510.

Edwards,R.A., Rodriguez-Brito,B., Wegley,L., Haynes,M., Breitbart,M., Peterson,D., Saar,M., Alexander,S., Alexander,E.C. and Rohwer,F. (2006) Using pyrosequencing to shed light on deep mine microbial ecology under extreme hydrogeologic conditions. *BMC Genomics*, **7**, 57.

Frishman,D., Mironov,A., Mewes,H. and Gelfand,M. (1998) Combining diverse evidence for gene recognition in completely sequenced bacterial genomes. *Nucleic Acids Res.*, **26**, 2941–2947.

Furrie,E. (2006) A molecular revolution in the study of intestinal microflora. *Gut*, **55**, 141–143.

Gouy,M. and Gautier (1982) Codon usage in bacteria: correlation with gene expressivity. *Nucleic Acids Res.*, **10**, 7055–7074.

Green,P. (1994) Documentation for PHRAP. http://www.genome.washington.edu/UWGC/analysistools/phrap.htm.

Lombardot,T., Kottman,R., Pfeffer,H., Richter,M., Teeling,H, Quast,C. and Gloeckner,F.O. (2006) Mex.net-database resources for marine ecological genomics. *Nucleic Acids Res.*, **34**, D390–D393.

Margulies,M., Egholm,M., Altman,W.E., Attiya,S., Bader,J.S., Bemben,L.A., Berka,J., Braverman,M.S., Chen,Y.J., Chen,Z., Dewell,S.B., Du,L., Fierro,J.M., Gomes,X.V., Godwin,B.C., He,W., Helgesen,S., Ho,C.H., Irzyk,G.P., Jando,S. C., Alenquer,M.L., Jarvie,T.P., Jirage,K.B., Kim,J.B., Knight,J.R., Lanza,J.R., Leamon,J.H., Lefkowitz,S.M., Lei,M., Li,J., Lohman,K.L., Lu,H., Makhijani,V.B., McDade,K.E., McKenna,M.P., Myers,E.W., Nickerson,E., Nobile,J.R., Plant,R., Puc,B.P., Ronan,M.T., Roth,G.T., Sarkis,G.J., Simons,J.F., Simpson,J.W., Srinivasan,M., Tartaro,K.R., Tomasz,A., Vogt,K.A., Volkmer,G.A., Wang,S.H., Wang,Y., Weiner,M.P., Yu,P., Begley,R.F. and Rothberg,J.M. (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**, 376–380.

Moore,J. E. and Lake,J. A. (2003) Gene structure prediction in syntenic DNA segments. *Nucleic Acids Res.*, **31**, 7271–7279.

Nekrutenko,A., Chung,W.Y. and Li,W.Y. (2003) An evolutionary approach reveals a high protein-coding capacity of the human genome. *Trends Genet*, **19**, 306–310.

Nekrutenko,A., Chung,W.Y. and Li,W.Y. (2003) ETOPE: evolutionary test of predicted exons. *Nucleic Acids Res.*, **31**, 3564–3567.

Ogata,H., Goto,S., Sato,K., Fujibuchi,W., Bono,H. and Kanehisa,M. (1999) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res*., **27**, 29–34.

Pruitt,K., Tatusova,T. and Maglott,R. (2005) NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res*., **33**, 501–504.

Schloss,P.D. and Handelsman,J. (2003) Biotechnological prospects from metagenomics. *Curr. Opin. Biotechnol*., **14**, 303–310.

Tringe,S. G. and Rubin,E. M. (2005) Metagenomics: DNA sequencing of environmental samples. *Nat Rev Genet*., **6**, 805–814.

Tyson,G.W., Chapman,J., Hugenholtz,P., Allen,E.E., Ram,R.J., Richardson,P.M., Solovyev,V.V, Rubin,E.M, Rokhsar,D.S. and Banfield,J.F. (2004) Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature*, **428**, 37–43.

Venter,J.C., Remington,K., Heidelberg,J.F., Halpern,A.L., Rusch,D., Eisen,J.A., Wu,D., Paulsen,I., Nelson,K., Nelson,W., Fouts,D.E., Levy,S., Knap,A.H., Lomas,M.W., Nealson,K., White,O., Peterson,J., Hoffman,J., Parsons,R., Baden-Tillson,H., Pfannkoch,C., Rogers,Y-H and Hamilton,S.O. (2004) Environmental genome shotgun sequencing of the sargasso sea. *Science*, **304**, 66–74.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e290–e297
doi:10.1093/bioinformatics/btl209

A combinatorial pattern discovery approach for the prediction of membrane dipping (re-entrant) loops

Gorka Lasso[1], John F. Antoniw[2] and Jonathan G.L. Mullins[1,*]

[1]Membrane Proteins Structural Bioinformatics Group, School of Medicine, Swansea University, Singleton Park, Swansea SA2 8PP, Wales, UK and [2]Wheat Pathogenesis Programme, Plant Pathogen Interactions Division, Rothamsted Research, Harpenden, Hertfordshire AL5 2JQ, England, UK

ABSTRACT

Motivation: Membrane dipping loops are sections of membrane proteins that reside in the membrane but do not traverse from one side to the other, rather they enter and leave the same side of the membrane. We applied a combinatorial pattern discovery approach to sets of sequences containing at least one characterised structure described as possessing a membrane dipping loop. Discovered patterns were found to be composed of residues whose biochemical role is known to be essential for function of the protein, thus validating our approach.

TMLOOP (http://membraneproteins.swan.ac.uk/TMLOOP) was implemented to predict membrane dipping loops in polytopic membrane proteins. TMLOOP applies discovered patterns as weighted predictive rules in a collective motif method (a variation of the single motif method), to avoid inherent limitations of single motif methods in detecting distantly related proteins. The collective motif method applies several, partially overlapping patterns, which pertain to the same sequence region, allowing proteins containing small variations to be detected. The approach achieved 92.4% accuracy in sensitivity and 100% reliability in specificity. TMLOOP was applied to the Swiss-Prot database, identifying 1392 confirmed membrane dipping loops, 75 plausible membrane dipping loops hitherto uncharacterised by topology prediction methods or experimental approaches and 128 false positives (8.0%).

Contact: j.g.l.mullins@swansea.ac.uk

1 INTRODUCTION

Membrane dipping loops

Polytopic membrane proteins are embedded membrane proteins composed of a bundle of α-helices that completely span the membrane. These transmembrane α-helices are generally connected by extramembrane loops of various lengths. However, crystallized structures of membrane proteins such as aquaporins or potassium channels have shown that membrane dipping loops (sometimes called re-entrant loops) can also interconnect α-helical transmembrane regions at the same side of the membrane. These loops are characterised by their particular structure: the N-terminal section of the loop partially transverses the lipid bilayer but with the C-terminal section then returning to the same side as the N-terminal section of the loop. It has been suggested that membrane dipping loops play major roles as selectivity filters in the aquaglyceroporin family (Gonen, *et al*., 2004; Harries, *et al*., 2004; Murata, *et al*., 2000; Ren, *et al*., 2001; Savage, *et al*., 2003; Stroud, *et al*., 2003; Sui, *et al*., 2001), potassium channels (Doyle, *et al*., 1998; Jiang, *et al*., 2002; Jiang, *et al*., 2003; Kuo, *et al*., 2003; Long, *et al*., 2005; Nishida and MacKinnon, 2002; Zhou, *et al*., 2001), chloride channels (Dutzler, *et al*., 2002; Dutzler, *et al*., 2003) and also act as gates of membrane pores, such as in the glutamate homolog transporter (Yernool, *et al*., 2004), and the protein conducting channel (Van den Berg, *et al*., 2004). Prediction of membrane dipping loops from protein sequence has proved difficult as such regions are frequently amphiphilic, containing hydrophobic sections that are too intermittent to be identified as membrane regions. Membrane dipping loops require interactions with adjacent highly hydrophobic helices to become inserted in the membrane and minimise the energy penalty imposed by location of polar or charge residues in a low dielectric environment. In-silico topology prediction approaches often fail to predict membrane dipping loops in polytopic α-helical membrane proteins due to their residue composition differing with that of membrane spanning segments. To date, the bioinformatics approaches of our group, working on the dipping loops of glycerol channels, in collaboration with Stefan Hohmann and colleagues, have relied upon homology modelling (Bill *et al*., 2001), and comparison of test sequences with those of known loops in terms of secondary structure and the propensity scoring of successive residues to reside in α or β conformation (Hedfalk *et al*., 2004; Karlgren *et al*., 2004; Tamas *et al*., 2003), underpinned by extensive laboratory work including measuring channel efflux, mutagenesis and genetic screening. Here we describe the development of a novel and reliable approach to the difficult problem of predicting dipping loops directly from sequence that may be generically applied to membrane proteins.

Pattern discovery

By evolution, conserved nucleotides and residues are often indicative of a common structural or functional role either at the gene or protein level. Sequence similarity detection methods have been successfully applied in fields such as gene discovery, splicing prediction, phylogenesis, protein structure and function prediction or gene expression analysis. Multiple sequence alignment techniques have become the routine approach to measuring sequence similarity and identifying important residues (Altschul, *et al*., 1990; Pearson

*To whom correspondence should be addressed.

and Lipman, 1988). These alignments can be used to develop different motif representation techniques such as single (Falquet, *et al*., 2002) or multiple motif methods (Attwood, *et al*., 1999; Henikoff, *et al*., 1999; Wu and Brutlag, 1995), profiles (Bucher, *et al*., 1996) and hidden markov models (Baldi, *et al*., 1994; Eddy, 1996; Krogh, *et al*., 1994). However, multiple sequence alignment methods have proved to be computationally very expensive (Wang and Jiang, 1994), and the accuracy of the alignment diminishes when distantly related sequences need to be aligned. An alternative approach was based on pattern discovery methods using an unaligned set of sequences. The problem of detecting all possible patterns in a set of sequences has also proven to be computationally expensive but heuristics and restrictions in the architecture of patterns (e.g. maximum length, number of non-wild elements) (Jonassen, *et al*., 1995; Rigoutsos and Floratos, 1998; Sagot, *et al*., 1995) have made it possible to analyse large set of biological sequences and discover structurally and functionally important patterns (Darzentas, *et al*., 2005).

We have applied a pattern discovery software, TEIRESIAS (Rigoutsos and Floratos, 1998), to various sets of protein subfamilies or families, depending on the residue conservation in the dipping loop region, where at least one of their members has been crystallized and its structure described in the PDB_TM database (Tusnady, *et al*., 2004) and/or in the literature as having at least one membrane dipping loop. The pattern discovery process is carried out using three different types of analysis: i) exact pattern discovery, ii) pattern discovery using a chemical equivalency set and iii) pattern discovery using a structural equivalency set. Our program, TMLOOP, uses the discovered patterns as weighted predictive rules to predict potential membrane dipping loops in polytopic membrane proteins. This software was used to explore the performance of a single motif method compared to a variation of this approach, called the collective motif method approach. Single motif methods require exact pattern matching to find structural or functional relatedness and therefore can miss distant relatives which contain small variations of the pattern (Scordis, *et al*., 1999). The collective method is based on the use of different patterns, partially overlapping, which belong to the same motif and therefore distant relative proteins containing small variations of the most common patterns can be co-detected.

2 METHODS

Data collection

Crystallized membrane proteins containing membrane dipping loops in their structure were identified in the PDB_TM database (Tusnady, *et al*., 2004; update 24/10/05). The predicted membrane dipping loops in each of the membrane proteins listed in the PDB_TM database were cross-referenced to the literature corresponding to the crystallized structures. Although these papers accurately describe the three-dimensional structure of membrane proteins, the boundaries of the lipid bilayer can only be approximated as membrane proteins need to be extracted from the membrane to elucidate their structure. Therefore, most of the loops predicted in the PDB_TM database as membrane dipping loops were found to be described but some loops were not identified in the literature and were considered as potential loops. Some structures contain additional membrane dipping loops that were not listed in the PDB_TM database and so these loops were also considered. In addition, a manual identification of membrane dipping loops in PDB structures of membrane proteins of known 3D structure (the Stephen White laboratory at the University of California, Irvine, http://blanco.biomol.uci.edu/Membrane_Proteins_xtal.html) was carried out to guarantee that all PDB structures containing a membrane dipping loop had been included. All identified membrane dipping loops were ultimately manually confirmed by being viewed in RasMol (Sayle and Bissel, 1992). In the PDB_TM database 50 structures containing membrane dipping loops and 69 membrane dipping loops were identified. The literature described 5 additional membrane dipping loops in 3 determined structures. No additional structures with membrane dipping loops were manually identified. Of the 50 PDB structures considered, 46 structures were used in this study as membrane proteins containing membrane dipping loops. Members of protein families covered by the crystallized structures containing dipping loops were obtained from the Swiss-Prot database (Boeckmann *et al*., 2003), regardless of their taxonomic group, using the Uniprot/Swiss-Prot family/domain classification. At this stage, the functional and structural annotation of proteins obtained from the Swiss-Prot database was analysed and entries with inappropriate or insufficient functional annotation were discarded from each set. In order to avoid redundancy, protein families were filtered based on the sequence identity of the members composing the set (Hobohm, *et al*., 1992). A bioinformatics tool, Non-Red (Liakopoulos and colleagues, Department of Cell Biology and Biophysics at the University of Athens, http://athina.biol.uoa.gr/bioinformatics/NON-RED/), was used to avoid redundant protein sequences in each set, by removing one of a pair of sequences with homology higher than a user-defined level. Here, Non-Red was used with a setting of the minimum alignment length to 80 and the minimum identity level to 95%. Therefore pairs of sequences sharing a sequence identity of 0.95 or higher were avoided by removing the protein sequence of the given pair more similar to the remaining proteins in the set. The filtered set was defined as the gold standard set for the study. Where the protein containing a dipping loop belonged to a particular subfamily, it was important to ascertain whether the structural motif was conserved only in that particular subfamily or instead was a common feature present in other subfamilies or in the entire protein family. ClustalW (Chenna, *et al*., 2003) was used to analyze the residue conservation in the sequence region pertaining to the dipping loop motifs across the entire protein family set. When no clear differences in residue conservation was observed between subfamilies it was taken that the membrane dipping loop was a structural motif conserved across the entire protein family. By contrast, when there was little or no conservation across the different protein subfamilies, loops were included in the pattern discovery process as members of the particular subfamily only, as there was no evidence that the given membrane dipping loop was conserved throughout the entire protein family.

Isolation of membrane dipping loop regions

For each crystallized protein containing one or more membrane dipping loops a set of similar proteins was assembled as described above. These sets were composed of membrane protein sequences that belonged to the same (sub)family as the crystallized membrane protein found in the PDB_TM database. However, there was no information in the corresponding Swiss-Prot file relating to the location of the membrane dipping loops. The determination of the location of these structural motifs in non-crystallized protein sequences was achieved by aligning the non-crystallized protein sequences, using ClustalW, against the sequence corresponding to the relevant crystallized membrane protein, also known as the reference sequence. The structural motif was then mapped onto the reference sequence and the equivalent motif located in the remaining sequences in the alignment. The beginning and end of each membrane dipping loop were obtained from the PDB_TM database and checked manually. In order to minimise potential errors in identifying the ends of each membrane dipping loop, or possibly missing the appropriate section, 5 residues before the predicted starting position and 5 residues after the predicted ending position were considered. Within each set, all sequences were then reduced to the region corresponding to the particular membrane dipping loop detected in the crystallized membrane protein. At this stage, for each membrane dipping loop detected, a set of partial sequences was assembled.

Pattern discovery using TEIRESIAS

The TEIRESIAS algorithm (Rigoutsos and Floratos, 1998) may be used to discover patterns in an unaligned set of nucleotide or amino acid sequences. This software performs unsupervised pattern discovery and reports maximal patterns without enumerating the entire solution. The algorithm restricts the pattern discovery process by limiting the search to patterns with user-defined parameters: the minimum number of literals in any pattern, the maximum extent of an elementary pattern and the minimum support required for a pattern (L, W and K respectively). For the purposes of these analyses L was set to 3 as it has been shown to be the minimum value for which the convolution stage successfully operates during the pattern discovery process (Rigoutsos and Floratos, 1998), W was set to the length of the structural motif to be analyzed in each set (normally between 20-30) in order to detect conserved pairs of residues located in different halves of the structural motif but that may be closely associated in 3D in the membrane, and K was set to the 70% of the sequences contained in each set. The pattern discovery process was carried out using three different types of discovery: i) exact (identical) pattern discovery, ii) pattern discovery using a chemical equivalency set and iii) pattern discovery using a structural equivalency set. Each set was analysed individually using TEIRESIAS, and the dipping loops considered in each set were classified into three different structural categories: helix-in-turn-loop-out, loop-in-turn-helix-out and helix-in-turn-helix-out. These sets were also clustered if sharing structural similarities or assembled from the same protein family and analysed together using TEIRESIAS to find common patterns in structurally related membrane dipping loop motifs and common patterns in membrane dipping loops possibly caused by ancestral gene duplication events.

Pattern validation

The patterns detected by TEIRESIAS were not in themselves guaranteed to be selective as it is not possible to include negative control sets in the pattern discovery process. Therefore, it may be possible to discover patterns from one particular set in other sets of membrane proteins, whose structure does not actually contain a dipping loop, leading to predictive rules with poor specificity. To validate the patterns, an additional tool was implemented, named PATTERNTEST, whose function was to validate the patterns obtained using TEIRESIAS against positive and negative control sets assembled by the user. The patterns discovered for each set were validated against protein sequences belonging to the remaining sets of membrane dipping loop motifs and against the negative control set composed of 363 membrane proteins known not to have membrane dipping loops in their structure. This set was assembled using sequences pertaining to crystallized membrane proteins whose structure was visually checked during the data collection process, and protein families contained in the Swiss-Prot database known not to have membrane dipping loops in their structures (e.g. GPCR family). The subsequent patterns discovered by TEIRESIAS, but found to be present in membrane proteins with a different dipping loop motif and/or membrane proteins without a dipping loop motif and/or in proteins with the corresponding dipping loop motif, but having the pattern outside this motif, were eliminated as candidate predictive rules for TMLOOP.

TMLOOP

A predictive tool was implemented, named TMLOOP, to predict membrane dipping loops in polytopic membrane proteins. TMLOOP uses patterns discovered by TEIRESIAS and validated by PATTERNTEST as weighted predictive rules where the weight was calculated by dividing the number of sequences in the training set containing a particular pattern by the total number of sequences in the training set. The software requires a set of user-defined parameters to run the prediction: i) I is the minimum inter-loop length required between two contiguous loops, where two different patterns would predict the same loop only if the distance of both matches in the sequence is lower than I; ii) S, the minimum pattern support, which restricts the patterns used for the prediction such that only the patterns whose support is equal or higher than S would be used as predictive rules; and iii) C, the minimum prediction confidence, which restricts the report of protein matches to those predictions with a score equal or higher than C.

TMLOOP was evaluated by tenfold cross-validation. During the evaluation process, the single motif approach, using the pattern with the highest support for each set, and the collective motif approach were compared and different values of I, S and C were tested to set up the optimum conditions to maximize the sensitivity and specificity of TMLOOP (Table 2, Figure 1).

Swiss-Prot database prediction

TMLOOP was applied to the Swiss-Prot database using the single motif method and the collective motif method (using values of I, S and C reporting the maximum predictive score during evaluation), a consensus prediction of membrane dipping loops was also undertaken (table 3). Predicted loops were classified as true positives, false positives or possible loops that may merit to be experimentally studied. In order to identify possible hitherto undesignated loops, it was required to identify structural or functional relatedness to the corresponding crystallized protein type known to have a similar membrane dipping loop. This was achieved by: i) searching for structural evidence of the loop or functional relatedness in Swiss-Prot annotation and/or in the IUBMB enzyme nomenclature database and/or in the TCDB transport classification database (Saier *et al*., 2006, http://www.tcdb.org/); ii) looking for distant relationships using BLASTP with an E-value cutoff of 100 (Darzentas, *et al*., 2005); iii) local residue conservation analysis using ClustalW; and iv) relative position of the predicted loop in sequence to the positions of the transmembrane regions in sequence.

3 RESULTS

Pattern discovery and validation of patterns

The 12 sets of partial sequences corresponding to membrane dipping loops found in potassium channels, secY/SEC61 alpha family, aquaglyceroporin family (two loops), sodium/dicarboxylate symporter family, ClC chloride channel family (four loops), psaF family and FecCD subfamily from the binding-protein-dependent family, were analysed using TEIRESIAS individually and combined as described above. Table 1 summarizes the pattern discovery analyses carried out and the subsequent validation of patterns. Only patterns whose support is $\geq$70% were collected.

TMLOOP evaluation by tenfold cross-validation

TMLOOP sensitivity and specificity was tested using different values of I, C and S. TMLOOP was also evaluated using predictive rules for the sole pattern with the highest support found for each training set. Table 2 and Figure 1 summarise the evaluation results (I was set to a default of 30, shown to be the most appropriate minimum inter-loop length, data not shown).

Prediction of membrane dipping loops in Swiss-Prot database

TMLOOP was used to predict membrane dipping loops in polytopic membrane proteins listed in the Swiss-Prot database (version 48.0). The database contained 194,317 protein entries where 29,127 were polytopic membrane proteins (15.0%). TMLOOP was run with two different sets of parameters: i) the single motif approach, using the individual pattern with the highest score for each membrane dipping loop analysed (I=30, while the parameters C and S were not relevant for this prediction; and ii) the collective method approach, using TMLOOP with the most optimal parameters of C and S obtained from the evaluation by tenfold cross-validation (I=30). The results are shown in Tables 3 and 4 (and supplementary information can be found at http://membraneproteins.swan.ac.uk/TMLOOP/Supplementary).

Table 1. Training sets and pattern discovery of membrane dipping loops

Gold standard sets		Pattern discovery and validation		Single motif method	
Membrane dipping loop set	No. of sequences	No. of patterns	No. of validated patterns	Top scoring pattern	Support
Helix-in-turn-loop-out K^+ channel	134	382, 103, 5	35, 10, 0	[ST]..[ST].G[FY]G	0.89
Helix-in-turn-loop-out secY/SEC61 alpha family	75	12, 0, 0	0, 0, 0	No patterns found	-
L1: Loop-in-turn-helix-out Aquaglycerolporin family	49	863, 73, 22	167, 21, 6	SG.H.N...[ST]	0.96
L2: Loop-in-turn-helix-out Aquaglycerolporin family	49	249, 32, 19	24, 1, 1	[ILMV]NP.R.....[ILMV]	0.94
Helix-in-turn-helix-out Binding protein dependent transport system permease family	25	7506, 479, 31	82, 43, 11	[AG].[ILMV].F[ILMV] [AG]L[IMV].P.[ILMV]	0.96
L1: Helix-in-turn-helix-out Cl^- channel family	35	936, 46, 14	29, 5, 3	[ILMV]G..GP.V	0.86
L2: Helix-in-turn-helix-out Cl^- channel family	35	2419, 97, 63	98, 35, 28	[AG].[AG].G[ILMV]... [FY].....[AG]..F..E	1.0
L3: Helix-in-turn-helix-out Cl^- channel family	35	610, 45, 10	9, 7, 2	P.G...P....G...G	0.91
L4: Helix-in-turn-helix-out Cl^- channel family	35	3751, 137, 26	66, 0, 0	[AG].....[ILMV]... [ILMV][ILMV]..E[ILMV]T	0.91
Helix-in-turn-helix-out psaF family	16	182, 41, 13	27, 16, 9	A.....G..WP..A	1.0
L1: Helix-in-turn-helix-out Na^+: dicarboxylate symporter family	46	3613, 347, 63	134, 19, 11	[ILMV].....T.S[ST]...[ILMV]P	0.89
L2: Helix-in-turn-loop-out Na^+: dicarboxylate symporter family	46	14887, 1327, 103	324, 142, 17	[ILMV].....[ILMV].....S.G..[AG][ILMV].... .[ILMV].[ILMV].....[ILMV]	0.96
L1: Loop-in-turn-helix-out L2: Loop-in-turn-helix-out Aquaglycerolporin family	98	333, 6, 5	27, 0, 0	[ST]G...NP[AG]	0.86
L1: Helix-in-turn-helix-out L3: Helix-in-turn-helix-out Cl^- channel family	70	106, 0, 0	0, 0, 0	No patterns found	-
L2 Helix-in-turn-helix-out L4: Helix-in-turn-helix-out Cl^- channel family	70	244, 0, 0	0, 0, 0	No patterns found	-
L1: Helix-in-turn-helix-out L2: Helix-in-turn-helix-out L3: Helix-in-turn-helix-out L4: Helix-in-turn-helix-out Cl^- channel family	140	0, 0, 0	0, 0, 0	No patterns found	-
L1: Helix-in-turn-helix-out L2: Helix-in-turn-loop-out Na^+: dicarboxylate symporter family	92	124, 2, 0	0, 0, 0	No patterns found	-
Helix-in-turn-loop-out loops	255	0, 1, 0	0, 0, 0	No patterns found	-
Helix-in-turn-helix-out loops	227	0, 0, 0	0, 0, 0	No patterns found	-
All dipping loops	565	0, 0, 0	0, 0, 0	No patterns found	-

Columns one and two describe the different sets of membrane dipping loops assembled. Columns three and four give the number of patterns obtained by chemical ([A,G], [E,D], [F,Y], [K,R], [I,L,M,V], [Q,N], [S,T]), structural equivalence ([C,S], [D,L,N], [E,Q], [F,H,W,Y], [I,T,V], [K,M,R]) and exact discovery respectively. No common patterns were observed between membrane dipping loops across the different structural categories, with the exception of L1 and L2 of the aquaglyceroporin family. Columns five and six describe the single motif method where column five lists those patterns found with the highest support, and column six shows the corresponding pattern support.

4 DISCUSSION

Referencing of discovered patterns

The patterns discovered by TEIRESIAS and validated with PATTERNTEST were considered in the light of the crystallographic structures and literature. These patterns were frequently found to belong to structural motifs, which were described as essential for the function of the protein. Furthermore the biochemical roles of several of the residues described in these patterns have been described in experimental studies, validating our approach. The dipping loop motifs found in potassium channels, aquaglyceroporins and loops 1 and 3 in ClC chloride channels have been described

Table 2. Evaluation of TMLOOP by tenfold cross-validation

				S			Top score pattern
				70	80	90	
C	0.01	Sensitivity	Av	95.64	92.87	43.24	87.05
			Sd	2.58	2.98	5.57	3.2
		Specificity	Av	95.93	98.18	98.88	100
			Sd	6.93	4.42	0.46	0
	0.1	Sensitivity	Av	90.57	**92.43**	43.09	87.05
			Sd	3.05	**2.48**	5.56	3.2
		Specificity	Av	95.25	**100**	100	100
			Sd	2.36	**0**	0	0
	0.3	Sensitivity	Av	85.93	87.09	41.36	87.05
			Sd	3.85	3.73	5.69	3.2
		Specificity	Av	100	100	100	100
			Sd	0	0	0	0
	0.5	Sensitivity	Av	77.35	84.76	41.07	87.05
			Sd	5.19	4.28	5.04	3.2
		Specificity	Av	100	100	100	100
			Sd	0	0	0	0
	0.7	Sensitivity	Av	63.55	76.78	38.45	87.05
			Sd	5.83	4.16	5.14	3.2
		Specificity	Av	100	100	100	100
			Sd	0	0	0	0

Two different approaches were carried out : i) using the pattern for each membrane dipping loop set with the highest score (top score approach, which is a single motif approach, in orange) using an I value of 30; and ii) using various values of S (minimum pattern support) and C (minimum prediction confidence) with a fixed I value (minimum inter-loop length) of 30 (collective motif approach, in blue). The results shown with the white background are the data relating to the optimal performance of TMLOOP.The top score approach, which proved to be a conservative approach, gave a confidence of 1.0 for each prediction since here TMLOOP uses just one rule per membrane dipping loop considered and therefore the prediction is based upon exact single pattern matching (either yes or no).

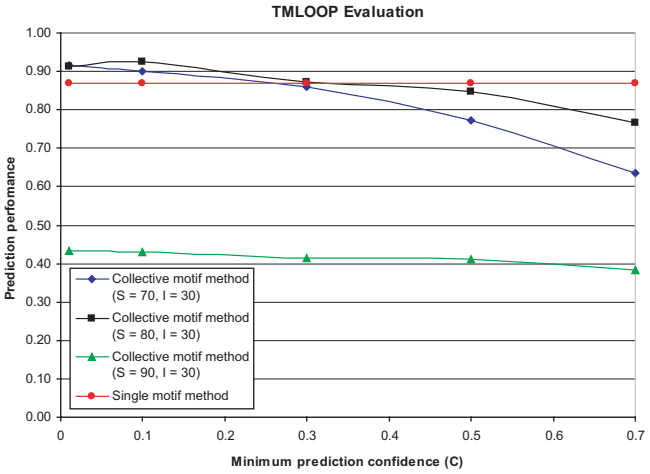


Fig. 1. Comparison of the performance of single and collective motif methods tested by tenfold cross-validation.

This graph shows the prediction performance (considering both the sensitivity and specificity) of each TMLOOP analysis (i. the single motif method in red, ii. the collective method –S (minimum pattern support) = 70, I (minimum inter-loop length) = 30- in blue, iii. the collective motif method –S = 80, I = 30- in black and iv. the collective motif method –S = 90, I = 30 in green) carried out at various levels of minimum prediction confidence (C). The collective method (S = 80, I = 30) showed the highest predictive score at a minimum confidence value C of 0.1. The C value of 0.3 is considered to be the threshold, below which the most accurate prediction method is the collective motif method and above which the single motif method performs better.

Table 3. Prediction of dipping loops in the SwissProt database

		True positives	False positives	Potential loops
Single motif method	Membrane dipping loops	1209	117	32
	Proteins	581	115	32
Collective motif method	Membrane dipping loops	1392	128	75
	Proteins	605	128	75
Consensus prediction	Membrane dipping loops	1204	78	31
	Proteins	576	78	31

The table summarises the analysis of the SwissProt database using TMLOOP (a) when only the pattern with the highest support is used (single motif approach) and (b) when all patterns whose support is ≥80 are used and only predictions with score ≥0.1 are reported (collective motif approach). The I value (minimum inter-loop length) was set to 30 for both methods. The last two rows (in red) show the consensus prediction considering both approaches.

as selectivity filters. The residues contained in patterns belonging to these selectivity filters have been extensively described and the discovered patterns were found to be refined motifs of those already proposed (e.g. the GYGD motif in potassium channels and the NPA motif in the aquaglyceroporin family). Loops 2 and 4 in ClC chloride channels have been proposed to link the two repeated halves within each monomer and make contacts with each other at the interface between monomers (Estevez and Jentsch, 2002). However, the precise functional relevance of specific residues is not clear. The most common patterns found in our analysis had support of 1.0 and 0.91 for loops 2 and 4 respectively, indicating important roles for particular residues.

In the case of the sodium / dicarboxylate symporter family both loops were proposed to act as gates in the membrane (Yernool, *et al*., 2004). The composition of patterns found in loop 1 were in agreement with the motifs identified in experimental studies, though a proline described previously as being conserved was not included in patterns relating to loop 2. This conserved proline is suggested to act as an anchor together with the serine-rich motif corresponding to loop 1. Further analyses showed that this proline was only conserved in 23 out of 46 sequences.

No patterns corresponding to discrete motifs could be found using TEIRESIAS for the detected dipping loops in the SecY/SEC61 alpha family, which transports soluble proteins across the membrane and passes membrane proteins into the membrane. The dipping loop found in this protein family is also known as the channel plug (Van den Berg, *et al*., 2004) and it has been suggested to block the pore in the closed state, in the open state the channel opens by displacement of the plug which moves away from the pore towards the plug-pocket (Collinson, 2005; Van den Berg, *et al*., 2004). Despite the overall importance of this motif no

Table 4. Newly predicted dipping loops in the SwissProt database

Swiss-Prot accession code	Definition	Predicted membrane dipping loop
Q9H2Y9	Solute carrier organic anion transporter family, member 5A1	helix-in-turn-helix-out ClC choride channel loop-1 like loop
Q8KWT2, Q8KWS7, P39642	Putative bacilysin exporter bacE	Loop-in-turn-helix-out Loop 1 & 2 aquaporin like
Q9NRA2, Q8BN82	Sialin (Solute carrier family 17 member 5)	helix-in-turn-helix-out ClC choride channel loop-1 like loop
Q58902	Hypothetical protein MJ1507	helix-in-turn-loop-out K^+ channel like
Q64SU9	Hypothetical transport protein BF2680	helix-in-turn-helix-out ClC choride channel loop-1 like loop
Q7UH36	Hypothetical transport protein RB4869	helix-in-turn-loop-out K^+ channel like
Q8AAG5	Hypothetical transport protein BT0500	helix-in-turn-helix-out ClC choride channel loop-1 like loop
Q57943	Hypothetical protein MJ0523	helix-in-turn-helix-out ClC choride channel loop-4 like loop
Q8NSS8	Hypothetical transport protein Cgl0590/cg0683	helix-in-turn-loop-out K^+ channel like
P74635	Hypothetical protein slr0753	helix-in-turn-helix-out ClC choride channel loop-1 like loop
P0AAC6, P0AAC7	Inner membrane protein yccA	helix-in-turn-helix-out Na^+: dicarboxylate symporter loop-1 like loop
P38745	Hypothetical 61.2 kDa protein in APM2-DUR3 intergenic region precursor	helix-in-turn-loop-out K^+ channel like
P37643	Inner membrane metabolite transport protein yhjE	helix-in-turn-loop-out K^+ channel like
P54181	Hypothetical protein ypnP	helix-in-turn-loop-out K^+ channel like
Q9V7S5	Putative inorganic phosphate cotransporter	helix-in-turn-helix-out ClC choride channel loop-1 like loop
P0A629, P0A628	Phosphate transport system permease protein pstC-1	helix-in-turn-helix-out ClC choride channel loop-1 like loop
P10603, P27182	ATP synthase C chain	helix-in-turn-helix-out ClC choride channel loop-1 like loop
P0A304, P0A305	ATP synthase C chain	helix-in-turn-loop-out K^+ channel like
Q8YGH4, Q8G1E6	Pyrophosphate-energized proton pump	helix-in-turn-loop-out K^+ channel like
P34299, Q8LGN0, Q9C5V5, O81078, Q9ULK0, Q61627, Q62640	Glutamate receptor precursor (glutamate-gated ion channel)	helix-in-turn-loop-out K^+ channel like
Q58671	Probable Na(+)/H(+) antiporter 3 (MjNapA)	helix-in-turn-loop-out K^+ channel like
Q15629, Q01685, Q15629, Q91V04, Q9GKZ4	Translocation associated membrane protein 1	helix-in-turn-loop-out K^+ channel like
Q8XED4, Q8FCT7, P33650, Q57IW8, Q5PLZ1, Q83ST5, P74884, Q57986, P73182	Ferrous iron transport protein B	helix-in-turn-loop-out K^+ channel like
Q97QP7, Q54875, Q59947, Q59986	Immunoglobulin A1 protease precursor	helix-in-turn-loop-out K^+ channel like
Q09917	Hypothetical protein C1F7.03 in chromosome I	helix-in-turn-loop-out K^+ channel like
Q8IZK6, Q8K595	Mucolipin-2	helix-in-turn-loop-out K^+ channel like
P91645, Q13936, Q01815, P15381, P22002, Q24270, Q01668, Q99244, P27732, O60840, Q02789, P07293, Q9JIS7, Q13698, O57483, Q02485, O73700, Q25452, P22316	Voltage-dependent calcium channel alpha-1 subunit	helix-in-turn-loop-out K^+ channel like
O28069 (top score pattern approach)	Hypothetical protein AF2214	helix-in-turn-helix-out ClC choride channel loop-4 like loop

A list of proteins, including the corresponding Swiss-Prot accession codes, containing plausible membrane dipping loops according to TMLOOP. Proteins listed were predicted by using either the single motif approach (I = 30) and/or the collective motif approach (S = 80, C = 0.1 and I = 30).

evidence of residue conservation was found in the multiple sequence alignment or in dipping loops in the pattern discovery process.

No experimental evidence has surfaced to describe the functional role of dipping loops belonging to the FeCD subfamily in the binding-protein-dependent permease family and in the PsaF family. However, the dipping loop region in the FeCD subfamily has been suggested to be important for binding the periplasmic binding protein BtuF (Locher, *et al*., 2002). The highest support found for a pattern corresponding to the membrane dipping loop in the FeCD subfamily (0.96) showed the importance of this motif for the function of the protein and supported the suggestion made by Locher *et al*. On the other hand, members of the PsaF family form part of the photosystem I (PSI). This family has been suggested

to mediate plastocyanin docking and fast electron transport kinetics in the eukaryotic PSI (Haldrup, *et al*., 2000; Hippler, *et al*., 1999). By contrast, in cyanobacteria PsaF proteins have been suggested to contribute to structural features on the surface of PSI and bind carotenoids which serve as a light harvesting and photo-protecting molecule (Jordan, *et al*., 2001). The highest support of patterns found in the dipping loop region (1.0) of PsaF proteins belonging to both cyanobacteria and eukaryote cells showed that this region was universally conserved across the taxa indicating potential residues with an essential and common functional role in both cyanobacteria and eukaryote cells.

TMLOOP evaluation

The main problem of single motif methods, is that prediction of a structural motif or functional category depends upon exact matching with a single pattern. Therefore distantly related proteins containing small variations of the pattern can not be detected. With TMLOOP, a single motif method (using the single pattern with the highest support found for each membrane dipping loop) can be employed to predict a particular membrane dipping loop, or instead, a set of partially overlapping patterns, may be used as weighted predictive rules (a collective motif method). The single motif approach and the collective motif approach using various combinations of C and S (I parameter was set to 30 in both approaches) were evaluated by tenfold cross-validation. The sensitivity and specificity of each method was calculated (table 2) and a single prediction performance score (a product of % sensitivity and % specificity divided by 10,000) was plotted against increasing minimum prediction confidence (C) values (Fig. 1).

Both methods performed well during the evaluation, however the single motif method approach was shown to be more accurate as C parameter increased. This is reflected in figure 1 where the C value of 0.3 is observed to be the threshold at which the accuracy of one method prevails over the other. When C values lower than 0.3 are considered the collective method is found to be the most accurate predictive method whereas when C values higher than 0.3 are considered the single motif method is the most accurate. The reason why the prediction accuracy of TMLOOP dropped significantly when S was set to 90 in the collective motif approach (table 2 and figure 1) was simply because some of the sets of patterns did not have a single pattern whose support was 0.90 or higher and therefore no patterns were considered for the prediction of the given membrane dipping loop. The evaluation showed that the collective approach (S = 80, C = 0.1, I = 30) was the most accurate method where TMLOOP achieved maximum values of sensitivity and specificity of 92.4% and 100% respectively (predictive score = 0.92, table 2 and figure 1). Although the single motif method was found to be a better approach with higher values of C, it also proved to be a conservative prediction. The flexibility of the collective motif approach allowed TMLOOP to detect 91.4% of the dipping loops contained in the two pore domain potassium channel family (in contrast to the 40.3% obtained by the single motif approach, data not shown), where each member of the family has been proposed to have two membrane dipping loops and the second loop showed small variations in sequence compared to the first dipping loop (successfully predicted by the single motif approach).These results reflect the strength of the collective motif method in being able to predict motifs similar but not identical to those used in the gold standard set.

This approach, where the dipping loop is specifically targeted, has distinct advantages over the baseline approach of identifying proteins that possess membrane dipping loops by ''association'' through global sequence similarity searching, where large portions of sequences may be common, but not the loop region, and *vice versa*. A thorough comparison of the targeted pattern approach with similarity search approaches is underway. This new approach has a further advantage in that it also predicts the specific residues composing the dipping loop, and the loop type. It is envisaged that the full value of TMLOOP will be realised through its use in conjunction with transmembrane region topology prediction programs.

Prediction of membrane dipping loops in the Swiss-Prot database

TMLOOP was applied to the Swiss-Prot database to predict membrane dipping loops in polytopic membrane proteins. Prediction was carried out by the two different approaches mentioned above: the single motif method approach using only the pattern with the highest support for each membrane dipping loop analyzed (I = 30); and the collective motif approach using TMLOOP with S, C and I set to 80, 0.1 and 30 respectively which maximized the predictive score during the tenfold cross-validation. The single motif method was shown to be a more conservative method whereas the collective motif method detected more potential membrane dipping loops not tested yet by experimental approaches (table 3). A good example of these highly plausible membrane dipping loops was found in the voltage-dependent calcium channel α-1 subunits (table 4) where a potassium-like membrane dipping loop (helix-in-turn-loop-out) was predicted (prediction score was 0.138). The low prediction score may be indicative of a distantly related structural motif that while not necessarily acting as a selectivity filter for potassium ions, may work for calcium ions in a similar fashion.

In conclusion, we have undertaken a full characterisation of all membrane dipping loops known to date. We have detected conserved patterns, with both high sensitivity and specificity, for most of these membrane dipping loop types. The corresponding literature highlighted some of the residues contained in these patterns as essential for the function of the protein, thus supporting our pattern discovery approach. We have implemented a tool to predict membrane dipping loops using a variation of the single motif method approach, named the collective motif approach, which was shown to be capable of detecting distantly related membrane dipping loops. Evaluation of TMLOOP by tenfold cross-validation showed impressive levels of both sensitivity and specificity. TMLOOP was successfully applied to the Swiss-Prot database predicting 75 plausible membrane dipping loops not detected previously by other methods. The program is available for use at http://membraneproteins.swan.ac.uk/TMLOOP (supplementary information can be found at http://membraneproteins.swan.ac.uk/TMLOOP/Supplementary).

ACKNOWLEDGEMENTS

We thank Vasilis Promponas for his valuable suggestions for identification of membrane dipping loops in crystallized membrane proteins. The work of G.L. was supported by a ''Beca de formación de investigadores'' grant from the Basque Government. Rothamsted

Research receives grant aided support from the Biotechnology and Biological Sciences Research Council (BBSRC) of the UK.

REFERENCES

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool, *J Mol Biol*, **215**, 403–410.

Attwood, T.K., Flower, D.R., Lewis, A.P., Mabey, J.E., Morgan, S.R., Scordis, P., Selley, J.N. and Wright, W. (1999) PRINTS prepares for the new millennium, *Nucleic Acids Res*, **27**, 220–225.

Baldi, P., Chauvin, Y., Hunkapiller, T. and McClure, M.A. (1994) Hidden Markov models of biological primary sequence information, *Proc Natl Acad Sci U S A*, **91**, 1059–1063.

Bill R.M., Hedfalk R., Karlgren S., Mullins J.G.L., Rydstrom J., Hohmann S. (2001) Analysis of the pore of the unusual MIP channel, yeast Fps1p. J. Biol. Chem. **276**, (39), 36543–36549.

Boeckmann B., Bairoch A., Apweiler R., Blatter M.-C., Estreicher A., Gasteiger E., Martin M.J., Michoud K., O'Donovan C., Phan I., Pilbout S., Schneider M. (2003) The Swiss-Prot protein knowledgebase and its supplement TrEMBL in 2003, *Nucleic Acids Res*. **31**:365–370.

Bucher, P., Karplus, K., Moeri, N. and Hofmann, K. (1996) A flexible motif search technique based on generalized profiles, *Comput Chem*, **20**, 3–23.

Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T.J., Higgins, D.G. and Thompson, J.D. (2003) Multiple sequence alignment with the Clustal series of programs, *Nucleic Acids Res*, **31**, 3497–3500.

Collinson, I. (2005) The structure of the bacterial protein translocation complex SecYEG, *Biochem Soc Trans*, **33**, 1225–1230.

Darzentas, N., Rigoutsos, I. and Ouzounis, C.A. (2005) Sensitive detection of sequence similarity using combinatorial pattern discovery: a challenging study of two distantly related protein families, *Proteins*, **61**, 926–937.

Doyle, D.A., Morais Cabral, J., Pfuetzner, R.A., Kuo, A., Gulbis, J.M., Cohen, S.L., Chait, B.T. and MacKinnon, R. (1998) The structure of the potassium channel: molecular basis of K+ conduction and selectivity, *Science*, **280**, 69–77.

Dutzler, R., Campbell, E.B., Cadene, M., Chait, B.T. and MacKinnon, R. (2002) X-ray structure of a ClC chloride channel at 3.0 A reveals the molecular basis of anion selectivity, *Nature*, **415**, 287–294.

Dutzler, R., Campbell, E.B. and MacKinnon, R. (2003) Gating the selectivity filter in ClC chloride channels, *Science*, **300**, 108–112.

Eddy, S.R. (1996) Hidden Markov models, *Curr Opin Struct Biol*, **6**, 361–365.

Estevez, R. and Jentsch, T.J. (2002) CLC chloride channels: correlating structure with function, *Curr Opin Struct Biol*, **12**, 531–539.

Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C.J., Hofmann, K. and Bairoch, A. (2002) The PROSITE database, its status in 2002, *Nucleic Acids Res*, **30**, 235–238.

Gonen, T., Sliz, P., Kistler, J., Cheng, Y. and Walz, T. (2004) Aquaporin-0 membrane junctions reveal the structure of a closed water pore, *Nature*, **429**, 193–197.

Haldrup, A., Simpson, D.J. and Scheller, H.V. (2000) Down-regulation of the PSI-F subunit of photosystem I (PSI) in Arabidopsis thaliana. The PSI-F subunit is essential for photoautotrophic growth and contributes to antenna function, *J Biol Chem*, **275**, 31211–31218.

Harries, W.E., Akhavan, D., Miercke, L.J., Khademi, S. and Stroud, R.M. (2004) The channel architecture of aquaporin 0 at a 2.2-A resolution, *Proc Natl Acad Sci U S A*, **101**, 14045–14050.

Hedfalk K, Bill R.M., Mullins J.G. L., Karlgren S, Filipsson C, Bergstrom J, Tamas M.J., Rydstrom J., Hohmann S. (2004) A regulatory domain in the C-terminal extension of the yeast glycerol channel Fps1p. J Biol Chem.**279** (15):14954–60.

Henikoff, J.G., Henikoff, S. and Pietrokovski, S. (1999) New features of the Blocks Database servers, *Nucleic Acids Res*, **27**, 226–228.

Hippler, M., Drepper, F., Rochaix, J.D. and Muhlenhoff, U. (1999) Insertion of the N-terminal part of PsaF from Chlamydomonas reinhardtii into photosystem I from Synechococcus elongatus enables efficient binding of algal plastocyanin and cytochrome c6, *J Biol Chem*, **274**, 4180–4188.

Hobohm, U., Scharf, M., Schneider, R. and Sander, C. (1992) Selection of representative protein data sets, *Protein Sci*, **1**, 409–417.

Jiang, Y., Lee, A., Chen, J., Cadene, M., Chait, B.T. and MacKinnon, R. (2002) Crystal structure and mechanism of a calcium-gated potassium channel, *Nature*, **417**, 515–522.

Jiang, Y., Lee, A., Chen, J., Ruta, V., Cadene, M., Chait, B.T. and MacKinnon, R. (2003) X-ray structure of a voltage-dependent K+ channel, *Nature*, **423**, 33–41.

Jonassen, I., Collins, J.F. and Higgins, D.G. (1995) Finding flexible patterns in unaligned protein sequences, *Protein Sci*, **4**, 1587–1595.

Jordan, P., Fromme, P., Witt, H.T., Klukas, O., Saenger, W. and Krauss, N. (2001) Three-dimensional structure of cyanobacterial photosystem I at 2.5 A resolution, *Nature*, **411**, 909–917.

Karlgren S., Philipson C., Mullins J.G.L., Bill R.M., Tamas M.J., Hohmann S. (2004) Identification of residues controlling transport through the yeast aquaglyceroporin Fps1 using a genetic screen. Eur J. Biochem **271**, 771–779.

Krogh, A., Brown, M., Mian, I.S., Sjolander, K. and Haussler, D. (1994) Hidden Markov models in computational biology. Applications to protein modeling, *J Mol Biol*, **235**, 1501–1531.

Kuo, A., Gulbis, J.M., Antcliff, J.F., Rahman, T., Lowe, E.D., Zimmer, J., Cuthbertson, J., Ashcroft, F.M., Ezaki, T. and Doyle, D.A. (2003) Crystal structure of the potassium channel KirBac1.1 in the closed state, *Science*, **300**, 1922–1926.

Locher, K.P., Lee, A.T. and Rees, D.C. (2002) The E. coli BtuCD structure: a framework for ABC transporter architecture and mechanism, *Science*, **296**, 1091–1098.

Long, S.B., Campbell, E.B. and Mackinnon, R. (2005) Crystal structure of a mammalian voltage-dependent Shaker family K+ channel, *Science*, **309**, 897–903.

Murata, K., Mitsuoka, K., Hirai, T., Walz, T., Agre, P., Heymann, J.B., Engel, A. and Fujiyoshi, Y. (2000) Structural determinants of water permeation through aquaporin-1, *Nature*, **407**, 599–605.

Nishida, M. and MacKinnon, R. (2002) Structural basis of inward rectification: cytoplasmic pore of the G protein-gated inward rectifier GIRK1 at 1.8 A resolution, *Cell*, **111**, 957–965.

Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence comparison, *Proc Natl Acad Sci U S A*, **85**, 2444–2448.

Ren, G., Reddy, V.S., Cheng, A., Melnyk, P. and Mitra, A.K. (2001) Visualization of a water-selective pore by electron crystallography in vitreous ice, *Proc Natl Acad Sci U S A*, **98**, 1398–1403.

Rigoutsos, I. and Floratos, A. (1998) Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm, *Bioinformatics*, **14**, 55–67.

Sagot, M.F., Viari, A. and Soldano, H. (1995) A distance-based block searching algorithm, *Proc Int Conf Intell Syst Mol Biol*, **3**, 322–331.

Saier, M.H. Jr, Tran, C.V., Barabote, R.D. (2006) TCDB: The transporter classification database for membrane transport protein analyses and information. *Nucleic Acids Res.*34:D181–6.

Savage, D.F., Egea, P.F., Robles-Colmenares, Y., O'Connell, J.D., 3rd and Stroud, R.M. (2003) Architecture and selectivity in aquaporins: 2.5 a X-ray structure of aquaporin Z, *PLoS Biol*, **1**, E72.

Sayle, R. and Bissel, A. (1992) RasMol: A Program for Fast Realistic Rendering of Molecular Structures with Shadows., *Proccedings of the 10th Eurographics UK 1992*. University of Edinburgh, UK.

Scordis, P., Flower, D.R. and Attwood, T.K. (1999) FingerPRINTScan: intelligent searching of the PRINTS motif database, *Bioinformatics*, **15**, 799–806.

Stroud, R.M., Miercke, L.J., O'Connell, J., Khademi, S., Lee, J.K., Remis, J., Harries, W., Robles, Y. and Akhavan, D. (2003) Glycerol facilitator GlpF and the associated aquaporin family of channels, *Curr Opin Struct Biol*, **13**, 424–431.

Sui, H., Han, B.G., Lee, J.K., Walian, P. and Jap, B.K. (2001) Structural basis of water-specific transport through the AQP1 water channel, *Nature*, **414**, 872–878.

Tamas M.J., Karlgren S., Bill R.M., Hedfalk K., Allegri L., Ferreira M., Thevelein J.M., Rydstrom J., Mullins J.G.L., Hohmann S. (2003) A short regulatory domain restricts glycerol transport through yeast Fps1p. J Biol Chem. **278** (8), 6337–45.

Tusnady, G.E., Dosztanyi, Z. and Simon, I. (2004) Transmembrane proteins in the Protein Data Bank: identification and classification, *Bioinformatics*, **20**, 2964–2972.

Van den Berg, B., Clemons, W.M., Jr., Collinson, I., Modis, Y., Hartmann, E., Harrison, S.C. and Rapoport, T.A. (2004) X-ray structure of a protein-conducting channel, *Nature*, **427**, 36–44.

Wang, L. and Jiang, T. (1994) On the complexity of multiple sequence alignment, *J Comput Biol*, **1**, 337–348.

Wu, T.D. and Brutlag, D.L. (1995) Identification of protein motifs using conserved amino acid properties and partitioning techniques, *Proc Int Conf Intell Syst Mol Biol*, **3**, 402–410.

Yernool, D., Boudker, O., Jin, Y. and Gouaux, E. (2004) Structure of a glutamate transporter homologue from Pyrococcus horikoshii, *Nature*, **431**, 811–818.

Zhou, Y., Morais-Cabral, J.H., Kaufman, A. and MacKinnon, R. (2001) Chemistry of ion coordination and hydration revealed by a K+ channel-Fab complex at 2.0 A resolution, *Nature*, **414**, 43–48.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e298–e306
doi:10.1093/bioinformatics/btl200

Interpreting anonymous DNA samples from mass disasters—probabilistic forensic inference using genetic markers

Tien-ho Lin[1], Eugene W. Myers[2] and Eric P. Xing[1,*]

[1]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA and [2]HHMI Janelia Farms Research Campus, Ashburn, VA

ABSTRACT

Motivation: The problem of identifying victims in a mass disaster using DNA fingerprints involves a scale of computation that requires efficient and accurate algorithms. In a typical scenario there are hundreds of samples taken from remains that must be matched to the pedigrees of the alleged victim's surviving relatives. Moreover the samples are often degraded due to heat and exposure. To develop a competent method for this type of forensic inference problem, the complicated quality issues of DNA typing need to be handled appropriately, the matches between every sample and every family must be considered, and the confidence of matches need to be provided.

Results: We present a unified probabilistic framework that efficiently clusters samples, conservatively eliminates implausible sample-pedigree pairings, and handles both degraded samples (missing values) and experimental errors in producing and/or reading a genotype. We present a method that confidently exclude forensically unambiguous sample-family matches from the large hypothesis space of candidate matches, based on posterior probabilistic inference. Due to the high confidentiality of disaster DNA data, simulation experiments are commonly performed and used here for validation. Our framework is shown to be robust to these errors at levels typical in real applications. Furthermore, the flexibility in the probabilistic models makes it possible to extend this framework to include other biological factors such as interdependent markers, mitochondrial sequences, and blood type.

Availability: The software and data sets are available from the authors upon request.

Contact: epxing@cs.cmu.edu

1 INTRODUCTION

Rapid advances in genotyping technology and mathematical theories of pedigrees have enabled their application in traditional forensic applications such as victim or perpetrator identification and paternity testing common place, even when family structures are complex or sample mixtures and mutations are involved (Mortera *et al*., 2003). A natural next step is to enlarge the scale of genetic forensic inference to mass disasters, such as airplane crashes, terrorist bombings, or battlefields, in which hundreds or even thousands of remains, usually highly degraded, have to be identified for all the victims according to DNA evidences from candidate family members (Egeland *et al*., 2000; Lauritzen and Sheehan, 2003). In addition to issues related to the increased scale of the problem, such a problem also poses new technical challenges such as the presence of errors in the genotypes and pedigrees, incomplete genetic information, and the need for decision making with very high confidence. (This last issue is typical of forensic cases, where seemingly low probability event such as incorrect victim/family matching can have serious legal consequence, and must be determined with a confidence much more stringent than usually adopted in experimental biology.)

DNA typing has long been used in forensic investigations, but until a decade ago, mass disaster victim identification has generally relied on dental and medical records, fingerprints, and even photographic evidence and personal effects (Ballantyne, 1997). These techniques require comparison between *ante mortem* (AM) information for the victim and *post mortem* (PM) information of the remains. However, in most mass disaster scenarios, AM information is not available for all victims and bodies are not intact, rendering such methods ineffective. Whitaker *et al*. (1995) established the use of short tandem repeat (STR) typing, or microsatellite markers, in mass disaster identification, and Olaisen *et al*. (1997) applied it to victim identification in the 1996 Spitsbergen aircraft accident, in which it proved to be highly reliable. A thirteen STR loci fingerprint set called the Combined DNA Index System (CODIS) is now in routine usage by the FBI, and has become a major tool in difficult disaster victim identification cases (Hsu *et al*., 1999; Cash *et al*., 2003).

While the basic problem of computing the likelihood ratio that a given sample is part of a given pedigree versus the null hypothesis of a random sample has been extensively studied (Olaisen *et al*., 1997), the inference problem of matching many pedigrees against many samples has not. Specialized software tools have been developed for large scale mass disaster identification (Cash *et al*., 2003) including the use of mitochondrial DNA (mtDNA) and single nucleotide polymorphism (SNP), but the matching algorithms utilized only rank the likely samples for each victim, and rank the likely victims for each sample. The complex interactions of all family evidence and all samples are not explored, and a great amount of expert involvement is still required. Moreover there is currently no systematic solution that addresses all the complicating factors: body part clustering, arbitrary pedigrees and their vetting, experimental genotyping error for the samples, partial genotypes due to heat and pressure damage of the DNA, and confidence of a cluster to family match based on other likely and

*To whom correspondence should be addressed.

unlikely family. This paper presents an architecture for the problem and a probabilistic framework that incorporates these uncertainties and scales to the required problem sizes.

We consider the following problem. We are given N family pedigrees for which the genotypes for some members are known, and the (potentially partial) genotypes of M samples belonging to the victims of a mass disaster. The problem is to match, with high confidence, the samples to the variable nodes (the purported victim reported by the family) of the pedigrees. Furthermore, we address how to screen out unambiguous matching outcomes and extract the truly ambiguous cases that merit costly personalized forensic investigation.

We approach the problem in two phases. First the samples are clustered into groups that have the same genotype. This reduces the problem of matching M samples to N pedigrees, to a smaller one of matching $J \ll M$ sample clusters to N pedigrees. During clustering possible errors in the STR data must be considered, especially when the DNA is degraded or when thousands of genotypes have been collected. We include a model for the types of errors that can occur in our probabilistic framework. In second phase, the cluster samples are matched to the variable nodes in the pedigrees. Forensic conclusions must be satisfactory from a legal perspective, as the purpose is to confirm the death of the victim, to return the remains to the families for closure, and in some cases to identify some of the victims as the perpetrators (in the case of terrorist acts). Therefore one can only make conclusions if there is a very small probability, typically 10^{-6} or smaller, of being wrong. We present a method to calculate the confidence of a certain match considering its likelihood ratio and other competitors for the slot. Then a forensically impossible match can be removed with high confidence.

Due to high confidentiality in disaster DNA data, simulation experiment is commonly performed so that true identity is known. We run three experiments with different simulation settings, and show that our algorithm is robust even with a lot of missing information and noise.

2 PRELIMINARIES

Consider M forensic samples from a mass disaster scene. Let $\mathbf{s}_1$, $\mathbf{s}_2$,...$\mathbf{s}_M$ denote the set of *sample genetic states* (to be specified shortly) retrieved from the M DNA samples, each from one of the forensic samples. Suppose there are N families that have filed missing person reports regarding this case (for presentation simplicity, we assume each family reports only one missing person, although generalization to multiple missing persons is feasible with our approach presented in the following), and have donated DNA samples as genetic references for victim identification. Let $\mathbf{f}_1$, $\mathbf{f}_2$ ... $\mathbf{f}_N$ denote the set of *familial genetic states* (defined in the sequel) obtained from these families.

Typically, body remains from a mass disaster and samples from donors are genetically characterized by a standard profile of K *microsatellite markers*. Each allele of such a marker corresponds to a numerical (in fact, discreet) reading from an electrophoresis gel; formally, we define each marker to be a random variable, and each of its alleles to be one of the realized states of this variable. For a forensic sample j, its sample genetic state (SGS) $\mathbf{s}_j \equiv (s_{j1}, s_{j2}, \ldots, s_{jK})$ denote the *genotype* profile of K markers, where $s_{jk} \equiv (s_{jk}^0, s_{jk}^1)$ represents an unordered pair of alleles of marker k from sample j. There are two alleles for each marker as human somatic cells are diploid, that is, there is a copy of a chromosome inherited from each parent. The superscripts ''1'' and ''0'' correspond to the parental origin of the alleles, i.e., paternal and maternal. Similarly, for each donor, we define $\mathbf{d}_j \equiv (d_{j1}, d_{j2}, \ldots, d_{jK})$ to be his/her genotype profile. Each family, say family i, may have multiple donors related by a *pedigree* T_i, therefore the familial genetic state (FGS) of a family with n_i donors is denoted by $\mathbf{f}_i \equiv \{\mathbf{d}_1, \ldots, \mathbf{d}_{n_i}; \mathbf{T}_i\}$. In typical mass disaster scenarios, multiple forensic samples (e.g., body remains) may belong to the same victim; therefore the samples can be grouped into clusters: i.e., $\mathbf{s}_1, \mathbf{s}_2, \ldots \mathbf{s}_M \Rightarrow \mathbf{c}_1, \mathbf{c}_2, \ldots \mathbf{c}_J$, where $\mathbf{c}_j = (\mathbf{c}_{j1}, \ldots, \mathbf{c}_{jm_j})$ and m_j denotes the size of cluster j (for simplicity, in the sequel we overload the symbol $\mathbf{c}_j$ to also represent the set of indices of SGSs belonging to cluster j). The forensic inference problem we concern here is that of determining the number of victims in the disaster, and the correct mapping between the victims and the reporting families.

In forensic applications, the microsatellite markers are chosen to be independent from each other (e.g., on different chromosomes). Via population censuring, the *a priori* probability (i.e., population frequency) of every allele of a microsatellite marker can be determined. Thus, given no familial information, the probability of an SGS of a forensic sample can be defined by the product of marker-specific genotype probabilities (by assuming the alleles are random samples from the population):

$$p(\mathbf{s}_j) = \prod_{k=1}^{K} p(s_{jk}), \tag{1}$$

where

$$p(s_{jk}) = \begin{cases} (\pi_{k,s_{jk}^0})^2 & \text{if} s_{jk}^0 = s_{jk}^1 \\ 2\pi_{k,s_{jk}^0}\pi_{k,s_{jk}^1} & \text{if} s_{jk}^0 \neq s_{jk}^1 \end{cases},$$

and $\pi_{k,a}$ denotes the population frequency of allele a of marker k.

The dependencies among donors from a family are captured by a pedigree. In our current setting, we consider only sexual inheritance among family members (i.e., donors plus the purported victim), and leave out nonsexual inheritance such as the mitochondria inheritance (incorporating such information is feasible in our framework and will be pursued in future research.) As illustrated in §3.4, a pedigree can be used to define the probability of the FGS of a family via a *probabilistic graphical model* (Pearl, 1988; Cowell *et al*., 1999). Note that a pedigree contains members who are not donors, nor victims, in order to specify the relations between the donors and the victim. These members represent the hidden variables in the graphical model, and will be marginalized out when computing the the FGS probability. For example, when the donor is the victim's brother, parents must appear on the pedigree even though their DNA samples are not available. The pedigree may have arbitrary structures, which are assumed to be correct after passing the validity check.

3 BODY IDENTIFICATION

To formulate a likelihood-ratio matching criteria for body identification, let's first assume that we have N reporting families and J victims (J will be determined by sample clustering as described in §3.2), and $J = N$. That is, each family has exactly one victim which corresponds to one cluster; and there is a one-to-one

alignment between the family pedigrees and the sample clusters. Our goal here is to find the optimal matching between $\{\mathbf{c}_j\}$ and $\{\mathbf{f}_i\}$. We will discuss how to relax the ''$J = N$'' and ''one-to-one correspondence'' assumptions later.

3.1 Matching via likelihood ratio

The matching between families and sample clusters can be represented by an $N \times N$ matching matrix $\mathbf{z}$, of which an element z_{ij} indicates the matching status between sample cluster j and family i:

$$z_{ij} = \begin{cases} 1 & \textit{if } \mathbf{c}_j \text{ is assigned to } \mathbf{f}_i \\ 0 & \text{otherwise} \end{cases}.$$

In case of one-to-one matching, $\mathbf{z}$ must satisfy the following constraints:

$$\sum_{i=1}^{N} z_{ij} = 1 \quad \forall j, \qquad \sum_{j=1}^{J} z_{ij} = 1 \quad \forall i. \tag{2}$$

Let $\pi(\mathbf{c}_j \mid \mathbf{f}_i)$ denotes the conditional probability of a cluster given a *matching* family, $\pi(\mathbf{c}_j)$ denotes the marginal probability of a cluster given no matching, and $p(\mathbf{f}_i)$ denotes the marginal probability of an FGS of family i. Assuming different families and different sample clusters are genetically independent given their matching configurations, the conditional probability of all FGSs $\{\mathbf{f}_j\}$ and clusters of SFSs $\{\mathbf{c}_j\}$, given the matching matrix $\mathbf{z}$, is:

$$\begin{aligned} p(\{\mathbf{c}_j\}, \{\mathbf{f}_i\} \mid \mathbf{z}) &= \prod_j p(\mathbf{c}_j \mid \{\mathbf{f}_i\}, \mathbf{z}) \prod_i p(\mathbf{f}_i) \\ &= \prod_{ij} \pi(\mathbf{c}_j \mid \mathbf{f}_i)^{z_{ij}} \prod_j \pi(\mathbf{c}_j)^{1-\Sigma_i z_{ij}} \prod_i p(\mathbf{f}_i) \\ &= \prod_{ij} \pi(\mathbf{c}_j \mid \mathbf{f}_i)^{z_{ij}} \prod_i p(\mathbf{f}_i). \end{aligned}$$

Note that according to the constraints of one-to-one matching in Eq. (2), we have $1 - \Sigma_i z_{ij} = 0$.

The likelihood ratio of an overall matching specification $\mathbf{z}$ versus a null hypothesis (that all families and samples are unrelated) is:

$$\begin{aligned} LR(\mathbf{z}) &= \frac{p(\{\mathbf{c}_j\}, \{\mathbf{f}_i\} \mid \mathbf{z})}{p(\{\mathbf{c}_j\}) p(\{\mathbf{f}_i\})} \\ &= \frac{\prod_j \prod_i \pi(\mathbf{c}_j \mid \mathbf{f}_i)^{z_{ij}}}{\prod_j \pi(\mathbf{c}_j)} \\ &= \prod_{ij} \left[\frac{\pi(\mathbf{c}_j \mid \mathbf{f}_i)}{\pi(\mathbf{c}_j)} \right]^{z_{ij}}. \end{aligned} \tag{3}$$

Let $\Lambda_{ij} \equiv \pi(\mathbf{c}_j \mid \mathbf{f}_i)/\pi(\mathbf{c}_j)$, and take the logarithm of LR, we have

$$\log LR(\mathbf{z}) = \sum_{j=1}^{J} \sum_{i=1}^{N} z_{ij} \log \Lambda_{ij}. \tag{4}$$

We postulate that an optimal body identification corresponds to a $\mathbf{z}$ that maximizes the likelihood ratio of matching family-clusters versus randomly generated $\{\mathbf{c}_j\}$ and $\{\mathbf{f}_i\}$. In the sequel we describe algorithms for identifying the sample clusters from the SGSs of samples, and for solving the optimal matching.

3.2 Sample clustering

The first problem in body identification is to determine the total number of victims in the case, and group body remains for each victim. We determine whether two samples, $\mathbf{s}_i$ and $\mathbf{s}_j$, are from the same victim or not based on the ratio of their joint probabilities under the two circumstances:

$$LR(\mathbf{s}_i, \mathbf{s}_j) = \frac{p(\mathbf{s}_i, \mathbf{s}_j)}{p(\mathbf{s}_i) p(\mathbf{s}_j)} = \frac{p(\mathbf{s}_i \mid \mathbf{s}_j)}{p(\mathbf{s}_i)} = \prod_{k=1}^{K} \frac{p(s_{ik} \mid s_{jk})}{p(s_{ik})}$$

The conditional probability $p(s_{ik} \mid s_{jk})$ of genotypes will be referred to as an *error model*, which will be specified in §3.2.2.

3.2.1 The union-find clustering algorithm Let each sample in the case be represented by a node, we can define an undirected graph over all samples of interest. Two nodes are connected if $LR(\mathbf{s}_i, \mathbf{s}_j) > \theta_c$, where θ_c is a user-specifiable threshold. As a common practice in mass disaster forensic identification, any two samples with more than two genotypes differences are immediately considered disconnected. Sample clustering is done by partition this graph into connected subgraph, which can be implemented efficiently using a *union-find algorithm*. We defines three operations: **make-set**—creates a set, **union**—merges two sets, and **find**—returns the host set of a node. The algorithm proceeds as follows:

(1) **make-set** creates a set for each node

(2) For two nodes of each edge, iterate the following

- **find** the corresponding sets,
- **union** the two sets (if they are connected by cross-set edges).

This process will converge to a clustering of samples, without a prior specification the number of clusters, but a threshold controlling the tightness of the clusters. This is a desirable feature in forensic inference because usually the legal agents would need to leverage their forensic experience and determine tolerable risk of legal decisions circumstantially. Once the clustering is complete, we extract a consensus SGS $\hat{\mathbf{c}}_j$ for each cluster $\mathbf{c}_j$ based on a maximum likelihood principle. That is, given the consensus $\hat{\mathbf{c}}_j$ that corresponds to the true genetic state (TGS) of a victim, the conditional probability of all SGSs of this cluster (i.e., this victim) is maximized:

$$\begin{aligned} \hat{\mathbf{c}}_j &= \arg \max_{\mathbf{t}} p(\mathbf{t}) \prod_{l \in \mathbf{c}_i} p(\mathbf{s}_l \mid \mathbf{t}) \\ &= \arg \max_{\mathbf{t}} \prod_{k=1}^{K} \Big(p(t_k) \prod_{l \in \mathbf{c}_j} p(s_{lk} \mid t_k) \Big), \end{aligned}$$

where the marker-specific conditional probability $p(s_{lk} \mid t_k)$ is given by the error model described bellow.

3.2.2 The error model The error model defines the probability distribution of a marker-specific sample genotype given the true genotype, $p(s_k \mid t_k)$. For two alleles $a \neq b$ of any markers (i.e., locus), we define five error types:

(1) Measurement error: Allele a is misread as $a \pm 0.1$ by the technician

(2) Calibration error: True genotype is (a, b) but calibration ladder is off by one, so instruments shows $(a+1, b+1)$ or $(a-1, b-1)$

(3) PCR Shutter error: True genotype is (a, a) but instruments shows $(a, a \pm 1)$

(4) Threshold error: True genotype is (a, b) but the b signal falls below threshold, so instruments shows (a, a)

(5) Mutation error: Allele a mutates to allele b

The probability of measurement, calibration, shutter, and threshold error are constants, denoted as ϵ_m, ϵ_c, ϵ_s, ϵ_t respectively. Based on the stepwise mutational model (Valdes *et al*., 1993) for microsatellite, the probability of a mutation from a to b is $p(b \mid a) = 0.5\mu(1 - \alpha)\alpha^{|b-a|-1}$, where μ is the mutation rate (probability of any mutation) and α is the factor by which mutation decreases as distance increases. Although this mutation distribution is not stationary (i.e. it does not ensure allele frequencies to be constant over the generations), it is simple and commonly used in forensic inference. Shutter, threshold, and calibration errors are defined on genotypes, but measurement and mutation errors are defined on alleles and have to consider two combinations, $p(s_k^0 \mid t_k^0)p(s_k^1 \mid t_k^1)$ and $p(s_k^0 \mid t_k^1)p(s_k^1 \mid t_k^0)$. To summarize, for $s_k \neq t_k$, we have:

$$p(s_k \mid t_k) = \begin{cases} \epsilon_c & \text{if } s_k^0 - t_k^0 = s_k^1 - t_k^1 = \pm 1 \\ \epsilon_s & \text{if } s_k^0 = s_k^1 = t_k^0,\ |s_k^1 - t_k^1| = 1 \\ \epsilon_t & \text{if } s_k^0 = t_k^0 = t_k^1 \\ \max(q(s_k^0; t_k^0)q(s_k^1; t_k^1), q(s_k^0; t_k^1)q(s_k^1; t_k^0)) & \text{otherwise} \end{cases},$$

where the allele error function $q(b; a)$ is defined as

$$q(b; a) = \begin{cases} 1 & \text{if } b = a \\ \epsilon_m & \text{if } |b - a| = 0.1 \\ 0.5\mu(1 - \alpha)\alpha^{|b-a|-1} & \text{otherwise} \end{cases}.$$

The $p(s_k \mid t_k)$ is a conditional probability that must sum to one. Thus, we define the "consistence" probability $p(s_k = t_k \mid t_k)$ as one minus all error probabilities, which is large comparing to the overall error probability (since the probabilities of each error type are always set to be very small):

$$p(s_k = t_k \mid t_k) = 1 - \sum_{s_k \neq t_k} p(s_k \mid t_k).$$

3.3 Pedigree inference

The conditional probability of a TGS given the FGS of a matching family, $p(\hat{\mathbf{c}}_j \mid \mathbf{f}_i)$, can be derived by pedigree inference. As discussed in Lauritzen and Sheehan (2003), the joint distribution of $\{\hat{\mathbf{c}}_j, \mathbf{f}_i\}$ defined by an arbitrary pedigree can be specified by a *probabilistic graphical model* (Pearl, 1988; Cowell *et al*., 1999), or more specifically, a *Bayesian network* (Pearl, 1986).

Recall that an FGS $\mathbf{f}_i$ is a two-tuple of donor genotypes $\{\mathbf{d}_1, \ldots, \mathbf{d}_{n_i}\}$ and a familial pedigree T_i. Based on T_i, we can construct a particular Bayesian network, known as *allele network*, or *gene pedigree* (Lauritzen and Sheehan, 2003), for all the alleles from all members (donor and non-donor) of the family and from the purported victim. Assuming that markers are independent and following the same pedigree, we construct an allele network for a single marker, say microsatellite k, as follows. For each individual, we introduce two allelic nodes, u_k^0 and u_k^1 (which are unobserved), denoting the maternal and paternal allele of this individual, respectively; and a genotype node u_k^g, which are observed for the donors and hidden for the non-donors in the family. Since the genotype is determined jointly by the two alleles, we have arcs pointing from each allelic node to its corresponding genotype node (Fig. 1 and Fig. 2). Due to Mendelian inheritance, the marker alleles in a decedent is dependent on that in his/her direct parents, thus we also have arcs pointing from the allelic nodes of a parent to the allelic nodes of the children. Note that the allelic nodes of individuals that are *founder* of the pedigree do not have any arcs pointing to them. For those individuals who are donors in a family (i.e., their genotype states are available from their DNA samples), we denote their corresponding genotype nodes as observed variables, shown as shaded circles. The genotype of the purported victim is also observed via sample clustering, but need to be matched correctly. In Fig. 1 and Fig. 2 we use circles with thick border to denote the genotype of a *candidate* victim. Because markers are independent in our case, each marker has a separate allele network with the same structure but different donor evidence (i.e., marker-specific genotypes). The joint probability of multiple markers is the product of all locus-specific marker probabilities defined by the allele network. Specifically, we use the following conditional distributions in our allele network model:

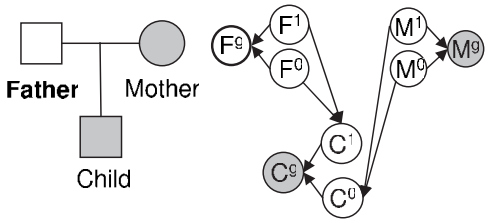


Fig. 1. A simple pedigree and its allele network, shaded nodes as donors and bold nodes as victim.

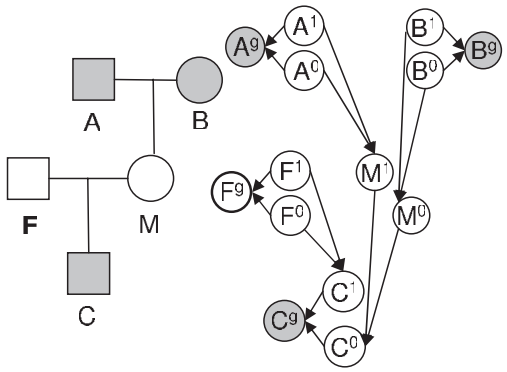


Fig. 2. A pedigree of three generations and its allele network.

(1) Founder distribution: $p(u_k^e) = \pi_{k,\,u_k^e}$, where $e \in \{0, 1\}$ represents the parental index of the allele, $\pi_{k,a}$ is the population frequency of allele a.

(2) Meiosis distribution: For an allele t_k^e inherited from a parent with genotype $s_k = \{u_k^0, u_k^1\}$, we have

$$p(t_k^e \mid u_k^0, u_k^1) = \begin{cases} 0.5 & \text{if } t_k^e = u_k^0 \text{ or } t_k^e = u_k^1, \text{ and } u_k^0 \neq u_k^1, \\ 1 & \text{if } t_k^e = u_k^0, \text{ and } u_k^0 = u_k^1, \\ 0 & \text{otherwise.} \end{cases}$$

(3) Genotype distribution: $p(u_k^g \mid u_k^0, u_k^1)$, which is specified by the error model defined in §3.2.2.

Given the allele network, and the above conditional distributions of a node in the network given its graph parents (not to be confused with biological parents), one can write down the joint distribution of all nodes, i.e. the victim and the FGS, as a product of all node-specific conditionals following a natural node ordering (e.g., from founder to decedents) (Pearl, 1988). From this joint probability we can derive conditional probability $p(x_F \mid x_E)$ of a set of variables $F \subsetneq V$ conditioned on a set of observed variables $E \subsetneq V$. F is called

query nodes, E is called *evidence nodes* and V is the totality of all nodes. The junction tree algorithm (Lauritzen and Spiegelhalter, 1988) can perform exact inference efficiently on a network of reasonable size, which is sufficient for our purpose.

3.4 Viterbi match: optimal body identification via linear programming

Given the conditional probabilities of TGSs of sample clusters and the FGSs of their matching families, $p(\hat{\mathbf{c}}_j \mid \mathbf{f}_i)$, now we are ready to tackle the optimal matching between sample clusters and families. Let us view the match matrix $\mathbf{z}$ as a representation of the edge configuration of a bipartite graph in which the clusters $\{\hat{\mathbf{c}}_j\}$ correspond to nodes in one partite, and the families $\{\hat{\mathbf{c}}_j\}$ correspond to the nodes in the other partite. Associating each edge between $\{\hat{\mathbf{c}}_j\}$ and $\mathbf{f}_i$ with a weight equal to $\log \pi(\hat{\mathbf{c}}_j \mid \mathbf{f}_i)/\pi(\hat{\mathbf{c}}_j)$, then the total cost of the matching, $LR(\mathbf{z})$, corresponds to the sum of weights of edges in the bipartite graph. Finding an optimal matching is equivalent to the classical maximum weight bipartite matching problem. We can solve this bipartite matching problem by mixed integer linear programming (LP):

$$\begin{aligned} &\max \sum_{j=1}^{J}\sum_{i=1}^{N} z_{ij} \log \Lambda_{ij} \\ &z_{ij} \in \{0,1\},\ \sum_{i=1}^{N} z_{ij} = 1\ \forall j,\ \sum_{j=1}^{J} z_{ij} = 1\ \forall i. \end{aligned} \tag{5}$$

There are many efficient algorithms and implementation for solving the above LP, and we use the open source Gnu Linear Programming Kit (GLPK) (Makhorin, 2001). Note that this approach gives a globally optimal mapping assignment between (equal number of) clusters and samples, analogous to finding the Viterbi path in hidden Markov model (but in this case an optimal matrix). Thus, we call the resulting body identification results a *Viterbi match*.

4 POSTERIOR MATCH AND MATCHING DISAMBIGUATION

The one-to-one constrain assumed so far in our algorithm is not always valid. In fact, since we cluster samples based on a tightness threshold rather than a given fixed number of clusters, we can not easily enforce $N = J$. In practice, a cluster may be unmatched, i.e. not assigned to any reporting family (e.g., due to poor sample quality, or nonexistence of the true claiming family); conversely, a family may also be unmatched (e.g., because no remain of the victim is found).

We assume each sample either comes from one family, or it is a random sample from the population. However, samples from one victim may be clustered into multiple clusters due to heterogeneity of the physical and measurement quality of different samples. To accommodate these flexibilities, we relax the normality constraints on the columns and rows of matching matrix $\mathbf{z}$, so that multiple clusters can be matched to one family, or no clusters or family get matched:

$$\sum_{i=1}^{N} z_{ij} \in \{0,1\}\ \forall j. \tag{6}$$

Furthermore, instead of seeking an overall estimate of $\mathbf{z}$, we would like to have a confidence measure of each of the judgments (i.e., match or not-match) specified by $\mathbf{z}$. From a forensic perspective, only matches with small enough probability should be considered (forensically) impossible, and excluded from legal consideration. In the sequel, we show how to calculate the posterior probability of a matching given cluster and family data; and then we show that, with this probability, how to screen out unambiguous matching outcomes and extract the truly ambiguous cases that merit costly personalized forensic investigation.

4.1 Posterior probability of a many-to-one matching

Now we derive the posterior probability of a matching given cluster TGSs and family FGSs, $p(\mathbf{z} \mid \{\mathbf{c}_j\}, \{\mathbf{f}_i\})$. According to the Bayes' theorem, we have:

$$p(\mathbf{z} \mid \{\mathbf{c}_j\}, \{\mathbf{f}_i\}) = \frac{p(\mathbf{z}) p(\{\mathbf{c}_j\}, \{\mathbf{f}_i\} \mid \mathbf{z})}{p(\{\mathbf{c}_j\}, \{\mathbf{f}_i\})}. \tag{7}$$

Since we do not know the matching *a priori*, $p(\mathbf{z})$ can be taken as uniform. Following the notations in §3.1, let $p(\mathbf{f}_i)$ and $\pi(\hat{\mathbf{c}}_j)$ denote the marginal probabilities of a given family, and a cluster TGS, respectively; and let $\pi(\hat{\mathbf{c}}_j \mid \mathbf{f}_i)$ denote the conditional probability a cluster TGS $\hat{\mathbf{c}}_j$ given its matching FGS $\mathbf{f}_i$ (i.e., $z_{ij} = 1$). Following the new constrain given by Eq. (6), and since the cluster TGSs are independent of each other given a matching $\mathbf{z}$, the conditional probability of each cluster TGS given a matching is:

$$p(\hat{\mathbf{c}}_j \mid \{\mathbf{f}_i\}, \mathbf{z}) = \begin{cases} \pi(\hat{\mathbf{c}}_j \mid \mathbf{f}_i) & \text{if } \exists i : z_{ij} = 1 \\ \pi(\hat{\mathbf{c}}_j) & \text{if } \sum_l z_{lj} = 0 \end{cases}, \tag{8}$$

Therefore the joint conditional probability of the TGSs and FGSs given $\mathbf{z}$ is

$$\begin{aligned} & p(\{\hat{\mathbf{c}}_j\}, \{\mathbf{f}_i\} \mid \mathbf{z}) \\ = \ & p(\{\hat{\mathbf{c}}_j\} \mid \{\mathbf{f}_i\}, \mathbf{z}) p(\{\mathbf{f}_i\} \mid \mathbf{z}) \\ = \ & \prod_j p(\hat{\mathbf{c}}_j \mid \{\mathbf{f}_i\}, \mathbf{z}) \prod_i p(\mathbf{f}_i) \\ = \ & \prod_{ij} \pi(\hat{\mathbf{c}}_j \mid \mathbf{f}_i)^{z_{ij}} \prod_j \pi(\hat{\mathbf{c}}_j)^{1-\sum_l z_{lj}} \prod_i p(\mathbf{f}_i) \\ = \ & \prod_{ij} \left[\frac{\pi(\hat{\mathbf{c}}_j \mid \mathbf{f}_i)}{\pi(\hat{\mathbf{c}}_j)}\right]^{z_{ij}} \prod_j \pi(\hat{\mathbf{c}}_j) \prod_i p(\mathbf{f}_i) \\ = \ & \prod_{ij} \Lambda_{ij}^{z_{ij}} \prod_j \pi(\hat{\mathbf{c}}_j) \prod_i p(\mathbf{f}_i). \end{aligned}$$

Thus, Eq. (7) reduces to:

$$p(\mathbf{z} \mid \{\mathbf{c}_j\}, \{\mathbf{f}_i\}) = \frac{1}{A} \prod_{ij} \Lambda_{ij}^{z_{ij}}, \tag{9}$$

where A is a normalizing constant summing over all $\mathbf{z}$. Using the fact that we are summing over all possible $\mathbf{z}$ under limitation (6), we can derive normalizing constant in closed form:

$$A = \sum_{\mathbf{z}} \prod_j \prod_i \Lambda_{ij}^{z_{ij}} = \prod_j (1 + \sum_i \Lambda_{ij}). \tag{10}$$

According to Eqs. (10) and (9), now we have a close-form expression for the posterior probability of a matching given the clusters and families data:

$$p(\mathbf{z} \mid \{\mathbf{c}_j\}, \{\mathbf{f}_i\}) = \frac{\prod_{ij} \Lambda_{ij}^{z_{ij}}}{\prod_j (1 + \sum_i \Lambda_{ij})}. \tag{11}$$

4.2 Individual posterior match and matching disambiguation

To qualify a candidate match, $\mathbf{c}_j$ versus $\mathbf{f}_i$, we compute the posterior probability of a match as follows. Let $\mathbb{Z}_{ij}$ denote the set of all matrix $\mathbf{z}$ in which $z_{ij} = 1$, i.e. all possible matching that assigns $\mathbf{c}_j$ to $\mathbf{f}_i$:

$$\mathbb{Z}_{ij} = \{\mathbf{z} : z_{ij} = 1\}, \tag{12}$$

Similarly, let $\mathbb{Z}_{ij}^c$ denote the complement of this set. Now the posterior probability of an *individual posterior match* (IPM) given TFSs of all samples clusters and FGSs of all reporting families can be computed as:

$$p(z_{ij} = 1 \mid \{\mathbf{c}_m\}, \{\mathbf{f}_l\}) = \sum_{z \in \mathbb{Z}_{ij}} p(\mathbf{z} \mid \{\mathbf{c}_m\}, \{\mathbf{f}_l\}) \tag{13}$$

To disqualify a candidate pair, $\mathbf{c}_j$ and $\mathbf{f}_i$, on the basis that they are extremely unlikely to be a true match, we define our *decoupling confidence* (DC) of this pair to be the posterior probability mass of the set $\mathbb{Z}_{ij}^c$, which can be computed as follows:

$$\begin{aligned}
& p(\mathbf{z} \in \mathbb{Z}_{ij}^c \mid \{\mathbf{c}_m\}, \{\mathbf{f}_l\}) \\
= \;& 1 - p(\mathbf{z} \in \mathbb{Z}_{ij} \mid \{\mathbf{c}_m\}, \{\mathbf{f}_l\}) \\
= \;& 1 - \sum_{\mathbf{z} \in \mathbb{Z}_{ij}} p(\mathbf{z} \mid \{\mathbf{c}_m\}, \{\mathbf{f}_l\}) \\
= \;& 1 - \sum_{\mathbf{z} \in \mathbb{Z}_{ij}} \tfrac{1}{A} \prod_m \prod_l \Lambda_{lm}^{z_{lm}} \\
= \;& 1 - \tfrac{1}{A} \Lambda_{ij} \prod_{m \neq j} \Big(1 + \sum_l \Lambda_{lm}\Big) \\
= \;& 1 - \frac{\Lambda_{ij} \prod_{m \neq j} \Big(1 + \sum_l \Lambda_{lm}\Big)}{\prod_m \Big(1 + \sum_l \Lambda_{lm}\Big)} \\
= \;& 1 - \frac{\Lambda_{ij}}{1 + \sum_l \Lambda_{lm}}.
\end{aligned}$$

Given the posterior probabilities of all IPMs, and the values of all DCs, now we can not only extract *maximum a posterior* (MAP) matches as in § 3, but also perform a *matching disambiguation* for the given $\{\mathbf{c}_m\}$ and $\{\mathbf{f}_l\}$. Essentially, for the later task we exclude a candidate match with DC higher than a specifiable threshold $1 - \theta_m$. Different values can be assigned to θ_m based on the situation of the disaster, and $\theta_m = 10^{-6}$ is commonly used in mass disaster scenes, meaning that by excluding the chosen pair of cluster TGS and family FGS, in less than one out of a million cases we missed a true match. If the DCs of all family-cluster pairs are higher than $1 - \theta_m$, then we are confident the cluster is unmatched, i.e. no family claims this victim.

After the aforementioned impossible-match exclusion, if there is zero or only one possible family for a cluster, this cluster is unambiguous and is considered determined. Otherwise, if a remaining cluster-family pair passes an IPM threshold, it is still considered a valid match. Finally, the clusters that still have ambiguity, i.e., with two or more possible families of IPM lower than the threshold, will be reported to human expert for further forensic investigate.

5 EXPERIMENTS

Due to high confidentiality of forensic DNA fingerprint data, a common practice in forensic science is to validate the models and algorithms via computer simulation experiments, for which the true matchings are known. Following convention, thirteen FBI CODIS markers are used. In each experiment we simulate N core families from a single population, by generating two random parents based on population allele frequencies, and generating one child from the parents. The victim is the child in three simulations, and in two other simulations the victim is one of the parents. Allele frequencies $\pi_{k,a}$ are assumed to be known and correct. Then we generate several TGSs for each victim, using the error model with different values of the parameters (to simulate different level of noise). The number of SGSs generated from a victim is distributed uniformly in an interval, $[M^{(0)}, M^{(1)}]$. Throughout the experiments, the parameters used for sample generation are intentionally set to be different from the ones used in our later inference, so that our test is unbiased and objective. For each marker, there is a probability of ϵ_u that the genotype is missing. The simulating parameter ϵ_u is set to be high, to represent that some samples are heavily degraded. However we require that the total number of available markers to be greater than 4 to make our cases forensically realistic—for situations where the recovered markers are less than or equal to 4, DNA evidence are usually dismissed due to lack of reliability. We performed five experiments with different simulating parameters, as described below:

(1) $N = 100$, $[M^{(0)}, M^{(1)}] = [3,7]$, so on average 500 samples. Victim is the child, and donors are the two parents. Simulation parameters are $\epsilon_u = 1/10$, $\epsilon_m = \epsilon_c = 0.001$, $\epsilon_s = \epsilon_t = 0.004$.

(2) A noisier setting, $N = 100$, $[M^{(0)}, M^{(1)}] = [3, 7]$, so on average 500 samples. Victim is one of the parents, and donors are the child and the other parent. Simulation parameters are $\epsilon_u = 1/4$, $\epsilon_m = \epsilon_c = 0.001$, $\epsilon_s = \epsilon_t = 0.004$.

(3) Similar to simulation 2 but with even more noise: $N = 100$, $[M^{(0)}, M^{(1)}] = [1, 9]$, so on average still 500 samples, but the cluster sizes vary more. The values of the simulation parameters are now higher, $\epsilon_u = 1/3$, $\epsilon_m = \epsilon_c = 0.002$, $\epsilon_s = \epsilon_t = 0.008$.

(4) Similar to simulation 1 but contains 500 families and on average 2500 samples (1,250,000 potential matches).

(5) Similar to simulation 1 but contains 1000 families and on average 5000 samples (5,000,000 potential matches).

The parameters used during computational inference in all four experiments are the same: $\epsilon_m = 0.00025$, $\epsilon_c = 0.00025$, $\epsilon_s = 0.001$, $\epsilon_t = 0.001$, which may be different from the parameters for sample simulation. The clustering LR threshold is $\theta_c = 500$. All experiments are repeated 9 times and their results are averaged.

5.1 Results on optimal body identification

Since our clustering is stringent, the number of resulting clusters is always greater or equal to the number of families ($N \leq J$), and the assumption of one-to-one mapping behind the Viterbi matching via LP no longer holds. We can still apply LP by enforcing the same optimization and constraint terms in Eq. (5), which means we still require one matching family for each cluster and one matching cluster for each family, but some clusters may be unmatched.

We perform optimal body identification using Viterbi matching via LP and MAP matching. We measure the performance by average false-negative rate (FN) and false-positive rate (FP), where FN is the ratio of undiscovered true matches to all true matches, and FP

Table 1. Optimal body identification performance of LP and MAP

Sim	LP FN	LP FP	MAP FN	MAP FP
1	0.0109	0.0	0.0	0.0
2	0.0130	0.0	0.0043	0.0043
3	0.0567	0.0112	0.0225	0.0225
4	0.0099	0.0004	0.0020	0.0020
5	0.0073	0.0002	0.0021	0.0021

Comparison of average false-negative (FN) and false-positive (FP) rate of LP and MAP algorithm. LP denotes the Viterbi match via LP based on one-to-one mapping assumption in § 3.4, and MAP denotes the MAP match based on many-to-one mapping in § 4.2.

is the ratio of incorrect predictions to all predictions. The results are shown in Table 1.

Overall, LP has low FP, but the FN is very high, mainly due to incorrectness of the one-to-one assumption in the model. MAP has slightly higher FP, but the FN is much lower. In simulation 1, MAP has zero FN and FP. Overall, both algorithms have good performance, even in the presence of noise and incomplete information. We are not aware of existence of any algorithm or software for this kind of forensic task in earlier and current literature.

5.2 Results on matching disambiguation

In a matching disambiguation task, our goal is to reduce as much as possible the amount of human effort in forensic inference by remove impossible cluster-family matches and high-confidence matches from a given mass disaster case. In this section, we compare the disambiguation results using the individual posterior match method with the ones using a conventional approach that excludes a candidate match by thresholding the likelihood ratio, e.g., a candidate match from $\mathbf{c}_j$ to $\mathbf{f}_i$ is excluded (i.e., deemed impossible) if $\Lambda_{ij} < \theta_m = 10^{-6}$. Such threshold means that the relative probability of a cluster-family match is only 10^{-6} compared to an alliterative hypothesis that they are unrelated.

We found that the accuracy of disambiguation via the posterior methods is significantly better than that of the conventional LR thresholding approach, as shown in Table 2. The threshold θ_m is set to be 10^{-6} in both algorithms. In our experiments, the accuracies are measured by: (1) the average percentage of remaining ambiguous clusters; (2) the average percentage of remaining ambiguous matching families for each cluster; and (3) the ratio of ambiguous family-cluster matches over all candidate matches. After applying the posterior match disambiguation algorithm, the remaining ambiguous clusters are almost always single samples. On average, the 500 samples were reduced to only 1, 5, and 13 ambiguous samples, in simulation 1, 2, and 3, respectively; and each ambiguous cluster has 6, 8, and 10 ambiguous candidate matching families, respectively. In simulation 4, 2500 samples and 500 families were reduced to 5 samples, each having 21 candidate families. In simulation 5, 5000 samples and 1000 families were reduced to 6 samples, each having 33 candidate families. Under the same noise level, larger sample size results in better reduction rate. The results of LR thresholding is generally much worse, about 3 to 12 fold increase in cluster ambiguity, and 3 to 5 fold increase in overall ambiguity.

Table 2. Comparison of disambiguation by posterior threshold and by LR threshold

Sim	Posterior Clusters	Posterior Families	Posterior Matches	LR thresholding Clusters	LR thresholding Families	LR thresholding Matches
1	0.01	0.06	0.0007	0.03	0.07	0.0019
2	0.04	0.08	0.0034	0.48	0.04	0.0190
3	0.12	0.10	0.0119	0.53	0.07	0.0371
4	0.01	0.04	0.0004	0.08	0.02	0.0013
5	0.01	0.03	0.0002	0.14	0.01	0.0010

Results of disambiguation by posterior and LR threshold. ''Clusters'' denote the average percentage of remaining ambiguous clusters. ''Families'' denote the percentage of ambiguous candidate matching families for each of these clusters. ''Matches'' denotes the ratio of ambiguous family-cluster matches over all possible matches. Parameter settings of the three simulations are described in 5.

A close examination of our results showed that these ambiguities all occurred in samples with severely degraded markers, typically with only 5 of the 13 marker readable. Under these circumstances, a family becomes a candidate match to a sample even when only 3 of the markers are compatible with that of the samples within an error range. In practice, such genetic samples would automatically be ruled legally insubstantiative even before computational forensic inference is conducted, and would require additional forensic evidence. Thus, our disambiguation results presented above is in fact a worst-case result, and the actual rate of disambiguation in real life can be much better if we are willing to insist on more stringent requirement for the quality of the DNA samples (e.g., by requiring more than half of the markers can be clearly typed). It is noteworthy that a domain expert does not need to examine the ambiguous families of each cluster one by one. An expert can determine the true family from evidences other than DNA, or determine the sample as unidentifiable, or repeat the DNA sampling.

5.3 Analysis of disambiguation threshold

The major difference between the posterior disambiguation and the LR-based method is that posterior disambiguation relates the LRs of all possible families versus a candidate cluster when inferring about each single matching. That is, for one cluster, if several likely matching families already exist, other families with lower LRs will be considered less likely, whereas in the conventional LR-based disambiguation, each candidate matching is assessed independent of other candidates. We illustrate this difference in disambiguation criteria in Figure 3. The histogram of all the log LR of simulation 1 and 2 is shown in Figure 3A and 3C. For the log LR of all possible families corresponding to a well-typed (i.e., with most markers measurable) cluster, as shown in Figure 3B and 3E, usually there are only a few (in this case, only one) candidate matches having LR above 10^{-6}, so the two methods make little (or no) difference because of nearly inexistence of between-match influences. However, for a degraded cluster illustrated in Figure 3C and 3F, there are many candidate matches with large LRs and they influence each other. Consequently the disambiguation via posterior inference tends to assess other candidates to be less likely than would have been suggested by the LRs alone. This effectively results in a criterion more stringent than 10^{-6}. The LR thresholding approach, on the other hand, still use the same threshold on LR. As shown in Figure 3C and 3F, the posterior match

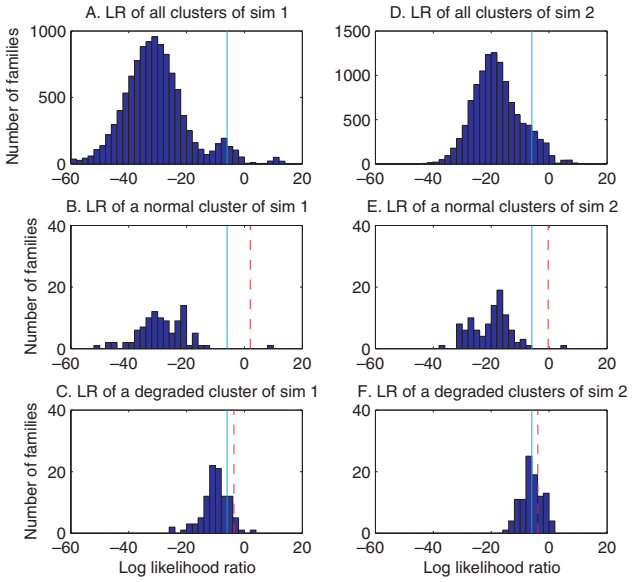


Fig. 3. The histogram of log likelihood ratio of simulation 1 and 2. **A–C** is based on simulation 1 and **D–F** is based on simulation 2. The x-axis is common logarithm of likelihood ratio, and the y-axis is number of families or matches. Vertical blue solid line denotes 10^{-6} threshold, and red dotted line denotes the effective threshold of disambiguation corresponding to the posterior match criteria. Specifically, we have: **A**. Distribution of all sample clusters of simulation 1. **B**. LR distribution of a well-typed cluster of simulation 1. **C**. LR distribution of a degraded cluster of simulation 1. **D–F**. The LR distributions of all sample clusters, a normal cluster, and a degraded cluster, respectively, in simulation 2.

method can reduces the ambiguity by a half or even more for degraded clusters.

In traditional forensic identification cases, which do not deal with DNA sample clustering but consider mostly high-quality anonymous samples, the LR of the correct identification tends to be very high, and there is usually no ambiguity. To see the difference in a mass disaster case, it is instructive to take a close look at the dataset and the ambiguous clusters and families reported by our algorithm. When there are fewer than 7 markers in a sample, typically there are indeed many ambiguous family pedigrees that cannot be excluded from a forensic perspective. For example, consider the highly degraded samples, of which an example is shown in Table 3. Typically such samples can have multiples plausible matching families, and the matches listed in Table 3 are only a few of all the likely matches. The ambiguity problem become very serious when the quality of the samples gets really poor, e.g., with fewer than 5 usable markers available. Essentially, the evidence become not enough for body identification—given only three or four markers, there could be too many perfect matches. In this case, the power of any computational and/or manual forensic inference diminishes, and we must seek additional evidence. We discuss some of the options in the next section.

6 DISCUSSION

Extending our probabilistic forensic inference methods to include other evidence is straightforward. For example, sometimes, in the forensic samples there also exist sequence data from the two segments of the hyper-variable control regions (e.g., regions 16,024 to 16,365 and 73 to 340) of the 16,569bp human mitochondria DNA (mtDNA). Because mtDNA has far more copies than the genome, they are often sequenceable when the genome is degraded and not sequenceable. Inheritance of mtDNA is maternal only, so there is much less uncertainty. But the mtDNA is less variable compared to microsatellites in genomic DNA. For example, while there are in principle 10 or more possible SNP differences in the mtDNA between any two individuals, a match is not conclusive due to high degeneracy of these polymorphism in human population. For example, about 7% of all Caucasian males have the same mtDNA sequence. Nevertheless, mtDNA can still be used to eliminate impossible matches, i.e., we can remove cluster-family matches with inconsistent mtDNA, and further reduce ambiguity.

Table 3. Case study of a highly degraded sample

Errors	Log LR	Description	THO1	D7S820	VWA
		Sample	(7,8)	(8,11)	(14,15)
0	1.70	True mate	(6,9)	(10,11)	(13,15)
		True child	(8,9)	(8,10)	(13,15)
0	1.00	Mate	(6,7)	(10,11)	(15,17)
		Child	(7,8)	(11,11)	(15,17)
1	−1.66	Mate	(7,9)	(9,10)	(18,18)
		Child	(6^{μ},9)	(10,11)	(15,18)
2	−4.12	Mate	(9,9.3)	(8,11)	(17,18)
		Child	(7,9)	($8,9^{s}$)	($18,18^{t}$)

A highly degraded sample of which three typed markers are shown. THO1, D7S820, and VWA are three markers in the CODIS system. The symbols a_s, a_t, a_{mn} denotes shutter, threshold, mutation error respectively. All the pedigrees have one of the parents as the victim and the other parent and a child as the donors. Among candidate families with high LR, four representative matches are listed here. Note that many different combinations are qualified for a match.

Occasionally, there will also be alleged direct sample evidence for a victim from a personal effect, such as a comb or tooth brush, in which case the genotype is available for the victim in the relevant family pedigree. Similarly, other factors like gender and blood type can be easily included using probabilistic rules.

In mass disaster scenes it is important to validate pedigree structure and donor evidence. For example, there may be an error in some donor's genotype, making it inconsistent with other donors' genotype. There is also the rather delicate issue that sometimes paternity or other blood relationships are not true. This kind of error can be detected by calculating the marginal probability of the evidence based on the allele network model. Families with probabilities under a threshold can be picked out and given to experts for examination. A family may have several victims in a mass disaster site. In this case one can introduce duplicated pedigrees one for each alleged victims. Each pedigree has the same structure and donor genotypes, but has different victim node. One must be careful about now the incorrectness of independence assumption for all pedigrees and for all the victim samples. For example, if a father and his son are both victims, their genotypes are not independent. This could slightly complicate the probabilistic inference computation for LR-based Viterbi match and posterior match.

Finally, it is noteworthy that, although in current forensic applications, genetic markers are usually chosen as independent

(e.g. the thirteen CODIS markers reside on different chromosomes), our probabilistic framework presented in this paper does not rely on the assumption that markers are independent. In extremely degraded disaster scenes, using single nucleotide polymorphism (SNP) for identification may be helpful (Cash *et al*., 2003); and for SNPs with high linkage disequilibrium, the markers are no longer independent. In such cases we can create an allele network with linkage probability, by adding a meiosis variable which couples different markers (Lauritzen and Sheehan, 2003). Under such circumstances, the allele network will become more complex and approximate inference or sampling may be necessary (Jordan *et al*., 1999; Xing *et al*., 2003).

In conclusion, we have presented a probabilistic modeling and inference framework for mass disaster victim identification. We expect that this framework can be easily generalized to handle more complicated forensic inference problems, and leverage richer forensic evidence or expert knowledge. It offers a promising platform to develop automatic expert system for a wide-range of forensic and genetic inference applications.

REFERENCES

Ballantyne,J. (1997) Mass disaster genetics. *Natural Genetics*, **15**, 329–331.

Cash,D.C., Hoyle,J.W. and Sutton,A.J. (2003) Development under extreme conditions: forensic bioinformatics in the wake of the World Trade Center disaster. In *Proceedings of Pacific Symposium on Biocomputing 2003,* **8**, 638–653.

Cowell,R.G., Dawed,A.P., Lauritzen,S.L. and Spiegelhalter,D.J. (1999) *Probabilistic Networks and Expert Systems*. Springer, New York.

Egeland,T., Mostad,P.F., Stenersen,M. and Mevag,B. (2000) Beyond traditional paternity and identification cases: selecting the most probable pedigree. *Forensic Science International*, **110**, 47–59.

Hsu,C.M., Huang,N.E., Tsai,L.C., Kao,L.G., Chao,C.H., Linacre,A. and Lee,J.C.-I. (1999) Identification of victims of the 1998 Taoyuan Airbus crash accident using DNA analysis. *International Journal of Legal Medicine*, **113**, 43–46.

Jordan,M.I., Ghahramani,Z., Jaakkola,T.S. and Saul,L.K. (1999) An introduction to variational methods for graphical models. *Learning in Graphical Models*. Kluwer Academic Publisher, pp. 105–161.

Lauritzen,S.L. and Spiegelhalter,D.J. (1988) Local computations with probabilities on graphical structures and their application to expert systems. *Journal of Royal Statistical Society (Series B)*, **50**, 157–224.

Lauritzen,S.L. and Sheehan,N.A. (2003) Graphical models for genetic analyses. *Statistical Science*, **18**, 489–514.

Makhorin,A (2001) *GNU Linear Programming Kit*. Moscow Aviation Institute, Moscow, Russia.

Mortera,J., Dawid,J. and Lauritzen,S.L. (2003) Probabilistic expert systems for DNA mixture profiling. *Theoretical Population Biology*, **63**, 191–205.

Olaisen,B., Stenersen,M. and Mevag,B. (1997) Identification by DNA analysis of the victims of the August 1996 Spitsbergen civil aircraft disaster. *Natural Genetics*, **15**, 402–405.

Pearl,J. (1986) Fusion, propagation and structuring in belief networks. *Artificial Intelligence*, **29**, 241–288.

Pearl,J. (1988) *Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA.

Valdes,A.M., Slatkin,M. and Freimert,N.B. (1993) Allele Frequencies at Microsatellite Loci: The Stepwise Mutation Model Revisited. *Genetics*, **133**, 737–749.

Whitaker,J.P., Clayton,T.M., Urquhart,A.J., Millican,E.S., Downes,T.J., Kimpton,C.P. and Gill,P. (1995) Short tandem repeat typing of bodies from a mass disaster: high success rate and characteristic amplification patterns in highly degraded samples. *BioTechniques*, **18**, 402–405.

Xing,E.P., Jordan,M.I. and Russell,S. (2003) A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the 19th Annual Conference on Uncertainty in AI*.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e307–e313
doi:10.1093/bioinformatics/btl226

Peptide sequence tag-based blind identification of post-translational modifications with point process model

Chunmei Liu[1,*], Bo Yan[2], Yinglei Song[1], Ying Xu[2] and Liming Cai[1,*]

[1]Department of Computer Science, University of Georgia, Athens, GA 30602 and [2]Department of Biochemistry and Molecular Biology, University of Georgia, Athens, GA 30602

ABSTRACT

An important but difficult problem in proteomics is the identification of post-translational modifications (PTMs) in a protein. In general, the process of PTM identification by aligning experimental spectra with theoretical spectra from peptides in a peptide database is very time consuming and may lead to high false positive rate. In this paper, we introduce a new approach that is both efficient and effective for blind PTM identification. Our work consists of the following phases. First, we develop a novel tree decomposition based algorithm that can efficiently generate peptide sequence tags (PSTs) from an *extended spectrum graph*. Sequence tags are selected from all maximum weighted antisymmetric paths in the graph and their reliabilities are evaluated with a score function. An efficient deterministic finite automaton (DFA) based model is then developed to search a peptide database for candidate peptides by using the generated sequence tags. Finally, a point process model—an efficient blind search approach for PTM identification, is applied to report the correct peptide and PTMs if there are any. Our tests on 2657 experimental tandem mass spectra and 2620 experimental spectra with one artificially added PTM show that, in addition to high efficiency, our *ab-initio* sequence tag selection algorithm achieves better or comparable accuracy to other approaches. Database search results show that the sequence tags of lengths 3 and 4 filter out more than 98.3% and 99.8% peptides respectively when applied to a yeast peptide database. With the dramatically reduced search space, the point process model achieves significant improvement in accuracy as well.

Availability: The software is available upon request.

Contact: {chunmei,cai}@cs.uga.edu

1 INTRODUCTION

It is a challenging problem to determine the amino acid sequence of a protein peptide from a tandem mass spectrum. The problem becomes more difficult when the spectrum contains post-translational modifications (PTMs). Existing computational methodologies for solving this problem can be classified into two major categories: database search based approaches and *de novo* peptide sequencing. Database search based tools such as SEQUEST (Eng *et al*., 1994) and Mascot (Perkins *et al*., 1999) compare a query spectrum with spectra from peptide sequences in a database and output those with high correlation scores as sequencing candidates. When the query spectrum contains PTMs, it becomes very difficult to select the correct peptide sequence since calculation becomes prohibitively slow, due to the enumeration and scoring of all possible modifications for each peptide from the database. In contrast, *de novo* sequencing methods (Chen *et al*., 2001; Dancik *et al*., 1999; Fernandez *et al*., 1995; Han *et al*., 2005; Hines *et al*., 1992; Liu *et al*., 2006; Ma *et al*., 2003; Searle *et al*., 2004; Taylor *et al*., 2001; Yan *et al*., 2005) aim to infer a peptide sequence from its spectrum directly without looking up a protein database. However, the accuracy of *de novo* sequencing is highly sensitive to the quality of the input spectrum. Usually it cannot infer a full length peptide sequence due to missing peaks, which consequently limits its application in practice.

Most of existing approaches (Perkins *et al*., 1999; Tanner *et al*., 2005; Wilkins *et al*., 1999; Yates *et al*., 1995) for identifying PTMs assume a limited set of modification types. These modification types can be modeled with pseudo amino acids; approaches developed for spectra free of PTMs can thus be directly applied to those with PTMs. However, spectra with unknown types of modifications may be erroneously processed with this method. Recently, a few approaches have been proposed for blind PTM identification (Tsur *et al*., 2005; Yan *et al*., 2006). In particular, (Tsur *et al*., 2005) proposes a dynamic programming algorithm to solve this problem. Alternatively, (Yan *et al*., 2006) introduces a point process model to process a spectrum, in which all possible optimal alignments between two spectra are obtained feasibly by computing the correlation of their corresponding processes. Both approaches are effective and able to detect unknown types of modifications. However, due to the large size of the search space, the optimal spectral alignment may be very time consuming and both approaches may suffer high false positive rate and computing inefficiency.

Recently, the idea of database filtration based on peptide sequence tags has been introduced to speed up peptide database search (Frank *et al*., 2005a; Tabb *et al*., 2003). For example, GutenTag (Tabb *et al*., 2003), which is based on a fragmentation model, generates many short sequence tags that are possibly contained in the peptide for a spectrum and compares the spectrum with ones from peptide sequences in a database that contain at least one of the selected tags. PepNovo (Frank *et al*., 2005a, b) evaluates the reliability of sequence tags on *de novo* sequencing results with a machine learning based approach and uses high reliable sequence tags to filter out most of the peptide sequences in a peptide database. With the reduced search space, the correct peptides can thus be

*To whom correspondence should be addressed.

identified efficiently with the conventional database search methods. Apparently, the methodology of combing *de novo* peptide sequencing and database search can dramatically improve the efficiency of peptide identification without sacrificing too much sensitivity. It thus may represent a promising approach for rapid and reliable peptide identification. However, the presence of PTMs significantly increases the difficulty of both *de novo* sequencing and database search. It is unclear what performance of these tools could be for generating correct peptide sequence tags and further finding out the correct PTMs through database search, in the presence of PTMs.

In this paper, we introduce an *ab initio* approach to sequence tag selection, which when further combines with the point process model (Yan *et al*., 2006) yields an efficient and accurate method for blind PTM identification. We have observed from our previous work (Liu *et al*., 2006) that the sequence tags can be selected from the maximum weighted antisymmetric path in a spectrum graph. Due to missing peaks or the shift of peaks in a spectrum that contains PTMs, a *de novo* sequencing algorithm may not be able to find a fully connected antisymmetric path that explains the spectrum. Nevertheless, it is possible to find all maximum weighted antisymmetric paths between certain pairs of vertices in the spectrum graph to obtain partial knowledge of the amino acid sequence of the spectrum. To efficiently implement this idea, we propose a novel tree decomposition based algorithm that can efficiently and effectively find all maximum weighted antisymmetric paths in a spectrum graph. We use the notion of *extended spectrum graph* that contains additional edges to describe the relationships between pairs of *complementary* vertices. Such a graph can deal with spectra with the presence of both b-ions and y-ions and ensure the antisymmetric property of the paths.

The algorithm has two major components. The fundamental component computes the maximum weighted antisymmetric paths connecting each pair of vertices contained in each tree node from a tree decomposition of the spectrum graph. Different tree decompositions are then generated from the fundamental component to find all maximum weighted antisymmetric paths between certain pairs of vertices. The time complexity of the algorithm is $O(6^t n(n+m))$, where t is the tree width of the tree decomposition and is usually small, n is the number of peaks in the spectrum, and m is the number of maximum weighted antisymmetric paths. Sequence tags (Frank *et al*., 2005a, b) are then selected from all maximum weighted antisymmetric paths and their reliabilities are evaluated with a score function.

We implemented our algorithm and applied it to PTM identifications. We first generated sequence tags from 2657 experimental yeast spectra downloaded from the Open Proteomics Database (OPD) (Prince *et al*., 2004). We compared the accuracy of the sequence tags with those generated by the popular tool PepNovo. Our experiments shows that our *ab initio* tag generation algorithm is significantly faster than PepNovo with comparable accuracies. We then manually added PTMs to 2620 spectra from the same data set and used our program to generate sequence tags and filter a yeast peptide database with a deterministic finite automaton (DFA) based model. The point process blind search model was then applied to the selected candidate peptides to identify the PTMs. Our experiments on the spectra with PTMs show that, compared with the results without database filtration, this combined approach can achieve significantly improved accuracy with 10 times and 80 times of speedups using the filtration of sequence tags of lengths 3 and 4 respectively.

2 MODELS AND ALGORITHMS

2.1 Extended spectrum graph and sequence tag selection problem

Although a spectrum may contain a few different types of ions, there are two mostly common ion types: N-terminal ions and C-terminal ions. For simplicity, we use b-ions and y-ions to represent them respectively. We assume $S = \{s_1, s_2, \ldots, s_m\}$ to be an experimental spectrum with complementary ions added if they are missing in the original experimental spectra. The possible mass values for the partial peptide for a peak s_i in the spectrum S form a set $V_i = \{s_i + \delta_1, s_i + \delta_2, \ldots, s_i + \delta_k\}$, where δ_k is the mass offset of ion i in the form of ion type k. Each of the mass values in V_i can be represented with a graph vertex and a vertex set $V = \{v_0\} \cup \cup_{i=1}^{m} V_i \cup \{v_n\}$ can thus be generated for S, where v_0 and u_n are two additional vertices with zero mass and the parent peptide mass respectively. A *spectrum graph* (Dancik *et al*., 1999) can be constructed upon V by connecting a directed edge from u to v if the mass difference between them is the mass of a single amino acid and the mass of u is less than that of v. u is an *in-neighbor* of vertex v and v is an *out-neighbor* of vertex u.

Based on a stochastic model for ions and peaks in a spectrum, vertices and edges in a spectrum graph can be assigned weights. Traditional approaches for *de novo* sequencing determine the amino acid sequence of a peptide by finding the maximum weighted path in the spectrum graph that connects v_0 and v_n. However, since a valid sequencing path only contains either b-ions or y-ions, it is necessary to identify pairs of vertices that cannot appear in the same sequencing path. A pair of vertices are *complementary* if a sequencing path can contain at most one of them. A path in a spectrum graph is *antisymmetric* if it contains at most one vertex from each pair of complementary vertices. A valid sequencing path is thus the maximum weighted antisymmetric path that connects v_0 and v_n. To address this issue, in addition to the directed edges in a spectrum graph, we also connect complementary vertices in the spectrum graph with undirected edges, yielding an *extended spectrum graph* (Liu *et al*., 2006). We show later in the paper that these undirected edges are important to ensure the antisymmetry of the paths found by our algorithm. Figure 1(a) through (c) show the spectrum of a short peptide and an *de novo* antisymmetric sequencing path contained in the corresponding extended spectrum graph.

For most of the spectra that contain PTMs, an antisymmetric path that connects v_0 and v_n may not exist in each of the corresponding spectrum graphs. As an example, Figure 1(b)(d) show a shift of peaks and the spectrum graph of the peptide with a PTM on one of its amino acids. However, we observe that parts of the amino acid sequence of the peptide can be obtained from maximum weighted antisymmetric paths between certain pairs of vertices. A path P in a spectrum graph is *maximum weighted ntisymmetric* if it satisfies the following constraints:

(1) P is antisymmetric,

(2) if u, v are the two ends of the path, any antisymmetric path P_1 that connects u and v has a weight no larger than that of P,

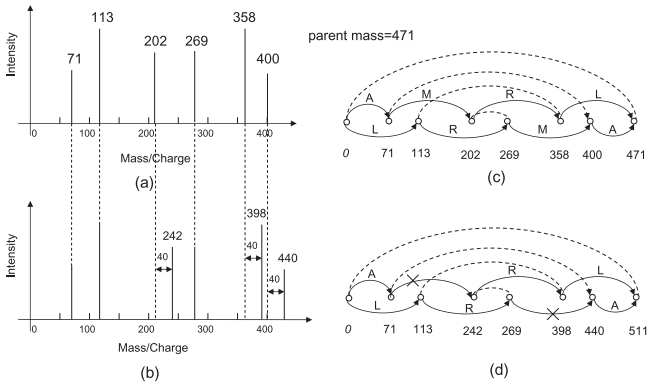


Fig. 1. (a) A tandem mass spectrum for a short peptide AMRL; for simplicity, only b and y-ions are included. (b) the spectrum for the same peptide, but with a PTM on amino acid M. (c) The extended spectrum graph for the spectrum in (a) and a longest antisymmetric sequencing path; dashed undirected edges connect complementary vertices. (d) The extended spectrum graph for the spectrum in (c), the original antisymmetric path for sequencing is disconnected due to the modification.

(3) there does not exist an antisymmetric path P_2 in the graph such that $P \subset P_2$.

2.2 Tree decompositions and path finding

DEFINITION 2.1 (Robertson *et al*., 1986) *Let $G = (V, E)$ be a graph, where V is the set of vertices in G, E denotes the set of edges in G. Pair (T, X) is a tree decomposition of graph italic G if it satisfies the following conditions*:

(1) *$T = (I, F)$ defines a tree, the sets of vertices and edges in T are I and F respectively*,

(2) *$X = \{X_i \mid i \in I, X_i \subseteq V\}$, and $\forall\, u \in V, \exists\, i \in I$ such that $u \in X_i$*,

(3) *$\forall (u, v) \in E, \exists\, i \in I$ such that $u \in X_i$ and $v \in X_i$*,

(4) *$\forall i, j, k \in I$, if k is on the path that connects i and j in tree T, then $X_i \cap X_j \subseteq X_k$.*

The tree width of the tree decomposition (T, X) is defined as $max_{i \in I} |X_i| - 1$. The tree width of the graph G is the minimum tree width over all possible tree decompositions of G.

As shown in Figure 2(a)(b), tree decomposition provides a new topological view on a graph. Based on a tree decomposition of a graph, many NP-hard optimization problems can be efficiently solved with a generic dynamic programming framework (Arnborg *et al*., 1989). In this framework, partial optimal solutions on subgraphs induced by vertices contained in subtrees can be extended and combined to obtain optimal solutions for larger subgraphs. In particular, partial optimal solutions can be combined with an exhaustive search performed only on vertices contained in a single tree node. The computation time needed by such a dynamic programming approach is thus dominantly determined by the tree width of the tree decomposition. Our testing results on 2657 experimental spectra show that the tree widths of extended spectrum graphs are generally around 5, which is sufficiently small for designing an efficient algorithm based on this framework.

The path-finding algorithm is based on the tree decompositions of the extended spectrum graph. The core part of the algorithm

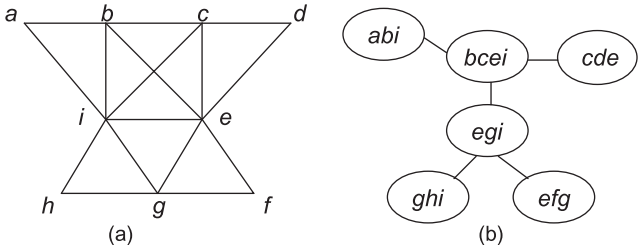


Fig. 2. (a) An example of a graph. (b) A tree decomposition for the graph in (a).

consists of two major components. In particular, in a given tree decomposition, the fundamental component finds the maximum weighted antisymmetric paths that connects each pair of the vertices contained in each tree node. To find all maximum weighted antisymmetric paths connecting certain pairs of vertices in the graph, the algorithm generates different tree decompositions and applies the procedure of the fundamental component to each of them. The overall time complexity of the algorithm is $O(6^t n(n+m))$, where t is the tree width, n is the number of vertices in the spectrum graph, and m is the number of maximum weighted antisymmetric paths in the spectrum graph.

2.2.1 The Fundamental Component The algorithm arbitrarily selects a tree node as the root of a tree decomposition and maintains a dynamic programming table for each tree node. It then proceeds from leaves of the tree to the root to fill in all the dynamic programming tables. The table for each tree node stores the weight of the partial maximum weighted antisymmetric path connecting each pair of vertices in the tree node.

For a tree node with t vertices, the dynamic programming table contains $2t + 1$ columns, of which the first t columns store the *selection* of each vertex in the node to form a subpath and the other $t - 1$ columns are used to store the *connection state* between each pair of consecutive selected vertices in the tree node. Two additional columns V and L store the *valid bit* and the maximum weight of the partial antisymmetric path associated with a combination of selections and connection states in the same table entry respectively.

The selection value of a vertex in a tree node is 1 if it is selected to be in the partial optimal path and 0 otherwise. The value of a connection state could be one of the integers in set $\{0, 1, \ldots, l\}$, where l is the number of children of the tree node. The connection state for a pair of consecutive selected vertices in the tree node is 0 if they are contiguous in the path and is i ($i > 0$) if the vertices on the path between the pair of vertices are covered by the subtree rooted at the ith child. The number of possible combinations of selections and connection states can thus be up to $(2(l + 1))^t$. However, since we can remove tree nodes with more than two children by generating extra tree nodes, the table for a tree node with t vertices may contain up to 6^t entries. The valid bit for a given entry is set to be 1 if there exists a partial antisymmetric path that follows the combination of selections and connection states in the entry.

To determine an entry in the table for a leaf node, the algorithm exhaustively enumerates and directly computes the validity and the maximum path weight for every possible combination of

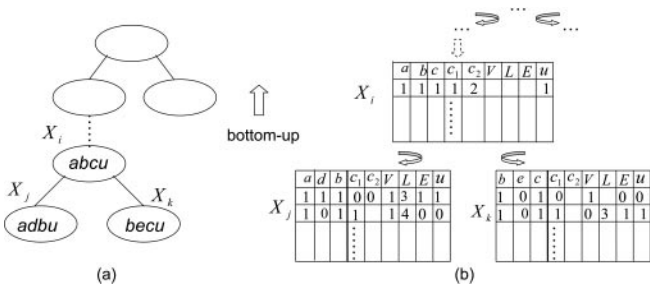


Fig. 3. A tree decomposition and its corresponding dynamic programming tables. The algorithm follows a bottom-up fashion starting with the leaf tree nodes. When computing the dynamic programming tables for an internal node X_i, the tables of its child nodes X_j and X_k need to be queried to compute the validity (V), the maximum path weight (L) and the extendibility (E) of a given entry in the table for X_i; vertex u is added to each tree bag to compute all the maximum weighted antisymmetric paths that start with it.

selections and connection states for vertices in the node. For an internal node, the algorithm refers to the tables of its children to determine the validity and the maximum path weight for each of its table entry. Figure 3 provides an example for computing the table entries for an internal node X_i. The computation time needed by the algorithm is $O(6^t n)$, where t is the tree width of the tree decomposition and n is the number of vertices in the graph. Due to space constraint, we refer the reader to our previous paper (Liu *et al*., 2006) for details.

2.2.2 Finding all maximum weighted antisymmetric paths The fundamental component can only compute the maximum antisymmetric paths between vertices that are included in at least one tree node. Further processing is thus needed to find all maximum weighted antisymmetric paths in the spectrum graph. For each vertex u in the spectrum graph, a new tree decomposition can be constructed by including u in all the tree nodes in the original tree decomposition. n different tree decompositions are thus generated.

To guarantee that the paths found by the algorithm satisfy the constraints of being maximum weighted antisymmetric, we further modify the previously described fundamental component. Specifically, as shown in Figure 3, one additional column E is added to each dynamic programming table to indicate whether the corresponding path can be extended to form an antisymmetric path with a larger weight by adding one of the in-neighbors of u to the path. This bit is set to be 1 if such an extension exists and 0 otherwise. The algorithm also sets the E bit to be 0 if the corresponding path for an entry does not start with u. This property for a given entry can be obtained by combining the E bits of its descendent entries with a direct inspection on vertices that are not included in the descendent entries. In view of the fact that the number of in-neighbors of u is bounded by 20, the aggregate computation time for this additional checking is $O(6^t n)$.

Based on the E bit of an entry, we are able to select the antisymmetric paths that start with u and cannot be extended with an in-neighbor of u. However, it is possible that some of the maximum antisymmetric paths can be extended from the other end of the path. We thus create an array S of size n and initialize all its elements to be zero. For any vertex v other than u in the spectrum graph, we can obtain $W(u, v)$, the weight of the maximum weighted antisymmetric path that connects u and v. For each out-neighbor v_t of v, we obtain $W(u, v, v_t)$, the weight of the maximum weighted antisymmetric path that passes through v and connects u and v_t. We then check whether $W(u, v, v_t)$ is equal to $W(u, v) + w(v, v_t)$ or not, where $w(v, v_t)$ is the weight of the edge (v, v_t). If it is the case for one out-neighbor of v, we set $S[v]$ to be 1, which suggests that the maximum antisymmetric path between u and v is extendable. The correctness of this operation is obvious since only in the case where the path is extendable, we can have one out-neighbor v_t of v such that $W(u, v, v_t) = W(u, v) + w(v, v_t)$. The aggregate time for this operation is again $O(6^t n)$. We then apply the tracing back procedure in the fundamental component to obtain all the maximum weighted antisymmetric paths starting with u. Based on the E bits and the array S, we can find all maximum weighted antisymmetric paths from the n tree decompositions and the total computation time is $O(6^t n(n + m))$, where m is the total number of maximum weighted antisymmetric paths. We thus have the following theorem.

THEOREM 2.1. *Given an extended spectrum graph $G = (V, E)$ and a tree decomposition of G with tree width t, all maximum weighted antisymmetric paths in G can be identified in time $O(6^t |V|(|V| + m))$, where m is the total number of maximum weighted antisymmetric paths in G.*

2.3 Reliability of sequence tags

We used the scoring scheme proposed in (Dancik *et al*., 1999) to assign weights to the vertices and edges in the extended spectrum graphs. The overall reliability of a sequence tag t_i was considered as a linear combination of normalized reliabilities $r_1(t_i)$ and $r_2(t_i)$ computed from the weights of the corresponding edges for t_i and an autocorrelation score developed in (Liu *et al*., 2005) respectively. In particular, the reliability $r(t_i)$ of sequence tag t_i is

$$r(t_i) = w_1 r_1(t_i) + w_2 r_2(t_i) \tag{1}$$

where $r_1(t_i)$ and $r_2(t_i)$ are computed with

$$r_1(t_i) = \frac{W(t_i)}{\sum_{l=1}^{q} W(t_l)} \tag{2}$$

$$r_2(t_i) = \frac{A(t_i)}{\sum_{l=1}^{q} A(t_l)} \tag{3}$$

where $W(t_i)$ is the sum of the weights of the edges that form t_i in the extended spectrum graph, q is the number of sequence tags, and $A(t_i)$ is an autocorrelation score computed with

$$A(t_i) = \sum_{k \in P(t_i)} I^*(k) I^*(n - k) \tag{4}$$

where $P(t_i)$ is the set of peaks that form t_i and $I*(k)$ and $I*(n - k)$ are adjusted intensities of complementary peaks k and $n - k$ in the spectrum. Both $r_1(t_i)$ and $r_2(t_i)$ are obtained by normalizing $W(t_i)$ and $A(t_i)$ over all sequence tags that are selected from the maximum weighted antisymmetric paths.

2.4 Database filtration with sequence tags

From the generated peptide sequence tags, we introduced a deterministic finite automaton (DFA) based model and used it to search a

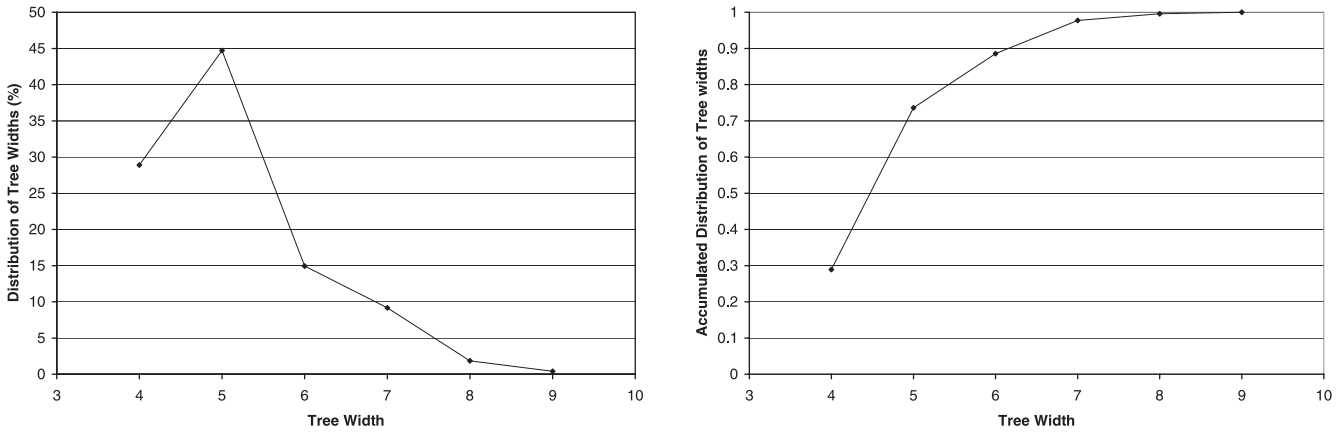


Fig. 4. The tree widths of the extended spectrum graphs for 2657 experimental spectra; left: the distribution of the tree widths, right: the cumulative distribution of the tree widths.

yeast peptide database which consists of 670,000 tryptically digested peptides (allowed up to 2 missing cleavages). Each amino acid in the tags represents a state of the DFA. We added an additional state as the start state. The accept states are the states that correspond to the last amino acids in the tags. Upon reading the first amino acid in a peptide sequence in the database, the DFA transfers from the start state to an appropriate state that corresponds to the first amino acid in a tag. The DFA then transfers from that state to next appropriate state upon reading the following amino acid in the peptide sequence. The procedure continues until the end of the peptide sequence. The peptide reading always goes forwards in the entire procedure. However, a trie based model (Frank *et al*., 2005a) needs to go back to the head of the trie each time when a substring of tag length in the peptide sequence has been examined. It thus may need more computation time than our DFA based model.

2.5 PTM identification by point process blind search

We finally apply the point process model for peptide identification and PTM search (Yan *et al*., 2006) on the candidate peptides after filtration. This model is an efficient blind search approach that does not require a list of pre-specified PTMs as input in advance. The algorithm attempts to find a set of optimal mass shifts to maximize the spectral alignment. Through one round of cross-correlation calculation, it is able to obtain all possible mass shifts feasibly (naturally this includes the optimal mass shifts). The computation time is independent of the number of PTMs, which outperforms most of the existing PTM identification tools whose computation time grows exponentially with the number of PTMs.

3 EXPERIMENTS AND DISCUSSIONS

3.1 Datasets

We downloaded 2657 annotated yeast ion trap tandem mass spectra from the Open Proteomics Database (OPD) (Prince *et al*., 2004). These spectra were selected based on the criteria with +2 precursor ion and $Xcorr \geq 2.5$ without PTMs. All the experimental mass spectra were ion trap data having a relative low mass resolution. We ran a data preprocessing procedure as described in (Frank *et al*., 2005b) to remove isotopic peaks and tiny noise peaks. Due to the shortage of reliably annotated spectra with PTMs in public domain, we constructed 2620 modified ones from those spectra by artificially adding one PTM from a common PTM pool to each spectrum. The detailed procedure is referred to (Yan *et al*., 2006).

3.2 Tree widths for spectrum graphs

Computing the optimal tree decomposition for a given graph is an NP-hard problem (Arnborg *et al*., 1987). A few efficient heuristics (Bodlaender, 1991) have been developed to compute a tree decomposition with small tree width for certain types of graphs. We used a greedy fill-in heuristic (Bodlaender, 1991) to find tree decompositions for the spectrum graphs of the experimental spectra. Figure 4 shows the distribution of the tree widths of the 2657 spectrum graphs. It can be clearly seen from the figure that the tree widths of about 90% of the spectrum graphs are bounded by 6, which are sufficiently small for developing an efficient tree decomposition based algorithm.

3.3 Sequence tag generation

We used our program to generate sequence tags at different lengths on the two datasets. We also ran the public available program PepNovo on the same datasets to obtain sequence tags. We then compared the generated tags with the sequencing results by SEQUEST and obtained the percentages of correct tags at different lengths for both of our program and PepNovo. Table 1 lists the results of our experiments on the two datasets and the comparison with PepNovo at different tag lengths. Our approach achieves comparable performance to PepNovo and is more computationally efficient (over 10 times faster than PepNovo at all different tag lengths). More importantly, our approach is *ab-initio* and does not require a training data set as PepNovo does. We believe further improvements in accuracy can be achieved if a more sophisticated model to evaluate the reliabilities of generated sequence tags is applied in the future.

3.4 Blind PTM identification by database search

After candidate peptides are filtered out with the sequence tags, our point process based blind search model is applied to evaluate these candidate peptides for further peptide identification and PTM detection. The results on 2620 modified spectra are listed in Table 2. It can be seen that the sequence tags of lengths 3 and 4 are able to

Table 1. A comparison between the performance of our tag selection program and that of PepNovo at different tag lengths

	Tag length	Algorithm	$r = 1(\%)$	$r = 3(\%)$	$r = 5(\%)$	$r = 10(\%)$	$r = 25(\%)$	T (s)
a	3	Our program	75.8	89.1	94.6	96.9	98.1	0.33
		PepNovo	75.8	90.1	93.6	96.8	98.8	3.62
	4	Our program	65.3	80.5	88.7	93.6	96.4	0.34
		PepNovo	65.5	81.0	86.6	92.3	95.3	3.69
	5	Our program	56.4	72.8	78.3	85.1	89.8	0.33
		PepNovo	58.4	71.3	77.6	84.0	88.9	3.83
	6	Our program	50.2	62.3	66.9	76.6	82.4	0.34
		PepNovo	49.7	61.5	67.8	75.0	81.8	4.27
b	3	Our program	68.1	84.8	90.3	94.8	97.1	0.32
		PepNovo	62.8	83.7	89.7	94.9	97.8	3.59
	4	Our program	53.5	71.2	78.6	84.8	90.0	0.32
		PepNovo	51.1	71.7	79.3	85.8	91.4	3.64

(a) on 2657 experimentalspectra without PTMs and (b) on 2620 experimentalspectra with one artificially added PTM. Columns for $r = 1, 3, 5, 10, 25$ represent the percentagesof spectra that have at least onecorrect tag in top 1, 3, 5, 10, 25 tags generated by our programand PepNovo respectively; T is the average time in seconds used for generating sequence tags for one spectrum.

Table 2. The accuracy of identifying PTMs from the modified spectra with selected tags of lengths 3 and 4

Tag length	Top 1	Top 2	Top 3	Top 4	Top 5	Filtration ratio	Time(s)
3	76.69	86.01	89.29	90.70	91.62	0.0167	263
4	74.98	80.77	81.71	82.17	84.40	0.0014	34
Without filtration	60.38	72.33	76.64	79.16	81.17	–	3843

The values at Top $i = 1, 2, 3, 4, 5$ represent the cumulative percentages of the search results capturing the original peptide sequences exactly in Top i; Filtration ratio is the ratio of the survived candidate peptides after tag filtration. Time is the total time in seconds used for the point-process blind search model to identify correct peptides and PTMs for all the 2620 experimental spectra. The last row is the results without sequence tag filtration.

filter out more than 98.3% and 99.8% peptides in the database respectively, which consequently speeds up the calculations dramatically. In addition, with the reduced search space and enriched signals of correct peptides, the accuracies of PTM identification by database search are significantly improved with both sequence tags of lengths 3 and 4. For example, with the filtration of tag length 3, approximately 77% and 92% of spectra are identified correctly as top 1 and within top 5 respectively, a significant improvement compared to the corresponding accuracies of 60% and 81% without database filtration. Increasing the tag length from 3 to 4 can further speed up the PTM identification by approximately 8 times. However, a slight drop in the identification accuracy is observed in this case due to the relative lower sensitivity of tag generation for tag length 4.

4 CONCLUSIONS

In this paper, we develop a novel tree decomposition based algorithm that can efficiently generate highly accurate sequence tags and conduct efficient PTM identification by combining sequence tag generation and database search. The algorithm models a spectrum with its corresponding extended spectrum graph and can find all maximum weighted antisymmetric paths in the spectrum graph with tree width t in time $O(6^t n(n+m))$, where n and m are the number of vertices and the number of maximum weighted antisymmetric paths in the graph, respectively. Sequence tags are then selected from all the maximum weighted antisymmetric paths. Our experiments show that this *ab-initio* approach can achieve accuracy comparable to that of PepNovo in a significantly reduced amount of computation time. More importantly, the sequence tags can be used to filter a peptide database effectively and thus enable the application of more accurate and sophisticated algorithms for PTM identification. In particular, we have built a rigid framework to conduct peptide identification and blind PTM search by combining high quality sequence tag generation and efficient database search. Experiments on spectra with PTMs show that this new approach can generate highly accurate sequence tags and significantly improve the accuracy of PTM identification by blind search.

ACKNOWLEDGEMENT

BY and YX's work was supported in part by National Science Foundation (NSF/DBI-0354771, NSF/ITR-IIS-0407204), and by a 'Distinguished Cancer Scholar' grant from the Georgia Cancer Coalition.

REFERENCES

Arnborg,S., Corneil,D. G. and Proskurowski,A. (1987) Complexity of finding embeddings in a *k*-tree. *SIAM Journal on Algebraic and Discrete Methods*, **8** (2), 277–284.

Arnborg,S. and Proskurowski,A. (1989) Linear time algorithms for NP-hard problems restricted to partial *k*-trees. *Discrete Applied Mathematics*, **23**, 11–24.

Bodlaender,H. L. (1991) Better algorithms for the pathwidth and treewidth of graphs. *Proceedings of the 18th international Colloquium on Automata, Languages and Programming*. Springer Verlag, Lecture Notes in Computer Science, **510,** 544–555.

Chen,T., Kao,M.Y., Tepel,M., Rush,J. and Church,G. M. (2001) A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry. *Journal of Computational Biology*, **8** (3), 325–337.

Dancík,V., Addona,T.A., Clauser,K.R., Vath,J.E. and Pevzner,P.A. (1999) *De Novo* Peptide Sequencing via Tandem Mass Spectrometry. *Journal of Computational Biology*, **6** (3/4), 327–342.

Eng,J.K., McCormack,A.L. and Yates III,J.R. (1994) An Approach to Correlate Tandem Mass Spectral Data of Peptides with Amino Acid Sequences in A Protein Database. *Journal of the American Society of Mass Spectrometry*, **5** (11), 976–989.

Ferńandez de Cossío,J., Gonzales,J. and Besada,V. (1995) A computer program to aid the sequencing of peptides in collision-activated decomposition experiments. *CABIOS*, **11** (4), 427–434.

Frank,A., Tanner,S. and Pevzner,P. (2005) Peptide Sequence Tags for Fast Database Search in Mass-Spectrometry. *Journal of Proteome Research*, **4** (4), 1287–1295.

Frank,A. and Pevzner,P. (2005) PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling. *Anal. Chem.*, **77** (4), 964–973.

Han,Y., Ma,B. and Zhang,K. (2005) SPIDER: Software for Protein Identification from Sequence Tags Containing Sequencing Error. *Journal of Bioinformatics and Computational Biology*, **3** (3), 697–716.

Hines,W.M., Falick,A.M., Burlingame,A.L. and Gibson,B.W. (1992) Pattern-based algorithm for peptide sequencing from tandem high energy collision-induced dissociation mass spectra. *J. Am. Soc. Mass. Spectrom.*, **3**, 326–336.

Liu,J., Ma,B. and Li,M. (2005) PRIMA: Peptide Robust Identification from MS/MS Spectra. *Proceedings of the Third Asia-Pacific Bioinformatics Conference*, 181–190.

Liu,C., Song,Y., Yan,B., Xu,Y. and Cai,L. (2006) Fast De Novo Peptide Sequencing and Spectral Alignment. *Proceedings of the Pacific Symposium on Biocomputing 2006*, 255–266.

Ma,B., Zhang,K., Hendrie,C., Liang,C., Li,M., Doherty-Kirby,A. and Lajoie,G. (2003) PEAKS: Powerful Software for Peptide De Novo Sequencing by Tandem Mass Spectrometry. *Rapid Communication in Mass Spectrometry*, **17** (20), 2337–2342.

Perkins,D.N., Pappin,D.J., Creasy,D.M. and Cottrell,J.S. (1999) Probability-based Protein Identification by Searching Sequence Databases Using Mass Spectrometry Data. *Electrophoresis*, **20** (18), 3551–3567.

Prince,J.T., Carlson,M.W., Wang,R., Lu,P. and Marcotte,E.M. (2004) The Need for a Public Proteomics Repository. *Nature Biotechnology*, **22** (4), 471–472.

Robertson,N. and Seymour,P.D. (1986) Graph Minors II. Algorithmic aspects of tree-width. *Journal of Algorithms*, **7**, 309–322.

Searle,B.C., Dasari,S., Turner,M., Reddy,A.P., Choi,D., Wilmarth,P.A., McCormack,A.L., David,L.L. and Nagalla,S.R. (2004) High-Throughput Identification of Proteins and Unanticipated Sequence Modifications Using a Mass-Based Alignment Algorithm for MS/MS De Novo Sequencing Results. *Anal. Chem*, **76** (8), 2220–30.

Tabb,D.L., Saraf,A. and Yates,J.R. (2003) GutenTag: High-Throughput Sequence Tagging via an Empirically Derived Fragmentation Model. *Analytical Chemistry*, **75**, 6415–6421.

Tanner,S., Shu,H., Frank,A., Wang,L.C., Zandi,E., Mumby,M., Pevzner,P.A. and Bafna,V. (2005) InsPecT: Identification of Posttranslationally Modified Peptides from Tandem Mass Spectra. *Analytical Chemistry*, **77** (14), 4626–4639.

Taylor,J.A. and Johnson,R.S. (2001) Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Anal.Chem.*, **73** (11), 2594–2604.

Tsur,D., Tanner,S., Zandi,E., Bafna,V. and Pevzner,P. (2005) Identification of Post-translational Modifications by Blind Search of Mass Spectra. *Nature Biotechnology*, **23** (12), 1562–1567.

Wilkins,M.R., Gasteiger,E., Gooley,A.A., Herbert,B.R., Molloy,M.P., Binz,P.A., Ou,K., Sanchez,J.C., Bairoch,A., Williams,K.L. and Hochstrasser,D.F. (1999) High-throughput Mass Spectrometric Discovery of Protein Post-Translational Modifications. *Journal of Molecular Biology*, **289** (3), 645–657.

Yan,B., Pan,C., Olman,V.N., Hettich,R.L. and Xu,Y. (2005) A Graph-Theoretic Approach for the Separation of b and y Ions in Tandem Mass Spectrometry. *Bioinformatics*, **21** (5), 563–574.

Yan,B., Zhou,T., Wang,P., Liu,Z., Emanuele II,V.A., Olman,V. and Xu,Y. (2006) A Point-Process Model for Rapid Identification of Post-Translational Modifications. *Proceedings of 2006 Pacific Symposium on Biocomputing*, 327–338.

Yates III,J.R., Eng,J.K. and McCormack,A.L. (1995) Mining Genomes: Correlating Tandem Mass Spectra of Modified and Unmodified Peptides to Sequences in Nucleotide Databases. *Analytical Chemistry*, **67** (18), 3202–3210.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e314–e322
doi:10.1093/bioinformatics/btl229

Identifying cycling genes by combining sequence homology and expression data

Yong Lu[1], Roni Rosenfeld[1] and Ziv Bar-Joseph[1,2,*]

[1]School of Computer Science and [2]Department of Biology, Carnegie Mellon University
5000 Forbes Ave. Pittsburgh, PA, 15213

ABSTRACT

Motivation: The expression of genes during the cell division process has now been studied in many different species. An important goal of these studies is to identify the set of cycling genes. To date, this was done independently for each of the species studied. Due to noise and other data analysis problems, accurately deriving a set of cycling genes from expression data is a hard problem. This is especially true for some of the multicellular organisms, including humans.

Results: Here we present the first algorithm that combines microarray expression data from multiple species for identifying cycling genes. Our algorithm represents genes from multiple species as nodes in a graph. Edges between genes represent sequence similarity. Starting with the measured expression values for each species we use Belief Propagation to determine a posterior score for genes. This posterior is used to determine a new set of cycling genes for each species.

We applied our algorithm to improve the identification of the set of cell cycle genes in budding yeast and humans. As we show, by incorporating sequence similarity information we were able to obtain a more accurate set of genes compared to methods that rely on expression data alone. Our method was especially successful for the human dataset indicating that it can use a high quality dataset from one species to overcome noise problems in another.

Availability: C implementation is available from the supporting website: http://www.cs.cmu.edu/~lyongu/pub/cellcycle/

Contact: zivbj@cs.cmu.edu

1 INTRODUCTION

The cell cycle system has been studied using microarray expression data in several species. These include humans (Whitfield *et al*., 2002), budding and fission yeast (Spellman *et al*., 1998; Rustici *et al*., 2004), plants (Menges *et al*., 2002) and bacteria (Laub *et al*., 2000). One of the first questions researchers face when analyzing such experiments is how to identify the set of cycling genes. Many methods have been developed for identifying such genes in a *single* species. These include methods that rely on Fourier transform (Spellman *et al*., 1998; Wichert *et al*., 2004), sinusoids (Schliep *et al*., 2003; Zhao *et al*., 2001), deconvolution (Bar-Joseph, 2004; Lu *et al*., 2004) and methods that combine expression amplitude with Fourier analysis (de Lichtenberg *et al*., 2005). All of the above methods rely on thresholds and other parameters which are not always easy to determine. Indeed, while these methods have been successful for some species, their success varied depending on the quality of the microarray data and the noise level (Shedden and Cooper, 2002).

The recent expression profiling of the cell cycle system in fission yeast provided a good opportunity for researchers to compare the set of cycling genes in two closely related species (budding and fission yeast). Surprisingly, the results indicated that cell cycle expression is not well conserved among these two species. As Rustici *et al*. (2004) write: ''*Our comparisons with budding yeast data revealed a surprisingly small core set of genes that are periodically expressed in both yeasts.*'' There could be many reasons for the disagreement between the list of cycling genes in different species. One possibility is that cell cycle expression is not well conserved (though cell cycle function may still be conserved on the post-transcriptional, or protein, level). However, there may be other reasons for this discrepancy. The different computational methods used to determine the set of cycling genes, noise in the data and differences in the quality of the data may result in one list being more accurate than the other. In such cases it may be possible to rely on one species to improve our detection of cycling genes in the other. This process may yield higher quality lists for both species.

In this paper we present a method for combining experiments from multiple species. Our algorithm combines sequence and expression data to identify the set of cycling genes. By considering sequence information we can use homologs to overcome noise and cutoff problems in individual species. By using expression data we can detect *functional* conservation, that is, sets of genes that are not only similar in sequence but also similar in function.

We use probabilistic graphical models, and in particular Markov random fields, to combine these data sources. We represent genes as nodes in the graph, with edges corresponding to sequence similarity as determined by a BLAST score. Each node (gene) is assigned an initial score which is determined by the expression experiment. Starting with this score we propagate information along the edges of the graph until convergence. Thus, if a node with a medium score is connected to a set of nodes with high scores, the information from the neighboring nodes can be used to elevate our belief in the assignment of this node, and vice versa.

Because the algorithm assumes expression conservation it leads to better agreement between cycling genes in different species. In order to test this algorithm it is thus important to show that

*To whom correspondence should be addressed.

this agreement does not come at the expense of a high quality set in either species. To show that our algorithm actually improves the quality of the identified set of cycling genes we tested it using two species for which additional information is available: Budding yeast and humans. As we show, by combining sequence and expression data our algorithm was able to detect a more accurate set of cycling genes in both species when compared to methods that rely on expression data only. While the improvement was mild for the high quality budding yeast expression data, it was much more substantial for the more noisy human cell cycle expression data.

1.1 Related work

Many methods have been suggested to identify the set of cycling genes from one or more expression datasets in a single species. For example, Spellman *et al*. (1998) used Fourier transform to identify cycling genes in budding yeast. Wichert *et al*. (2004) presented statistical methods for identifying periodically expressed genes and applied them (separately) to human and yeast. Lu *et al*. (2004) and Bar-Joseph *et al*. (2004) presented methods for deconvolving yeast expression data in order to improve the identification of cycling genes. de Lichtenberg *et al*. (2005) used scores that look at both, the amplitude of the expression value peak as well as the peak in the Fourier spectrum around the cell cycle period. Unlike the above methods, our method combines information from multiple species using sequence similarity. This allows us to overcome noise and improve the identification of cycling genes.

A number of previous papers combined sequence and expression data to study similarities in expression between different species. For example, Bergmann *et al*. (2004) clustered data from six different species to identify modules of genes that are co-expressed. Stuart *et al*. (2003) identified 'metagenes', a group of homolog genes from four different species (one gene from each species), and then used correlation coefficients to link metagenes forming a co-expression network. Our work differs from these papers in several important aspects. First, unlike prior work that relied on clustering to identify groups of co-expressed genes under a wide range of conditions, our approach uses a *classification* framework to achieve a different goal: identifying a set of conserved cycling genes. Second, prior work only looked at pairwise expression similarities, whereas our algorithm utilizes the complete graph topology to propagate information. Finally, previous papers used sequence similarity as a binary value (similar or not). In contrast, our framework uses the extent of this similarity to determine edge weights. The higher the similarity the greater the importance of neighboring genes for determining the cyclic score.

Recently, a number of papers compared the regulatory networks of various species (Sharan *et al*., 2005). These papers used graph theoretic methods to compare networks across species and identify similar pathways in these species. The focus of these papers and their goals are very different from ours. While we are focused on identifying the set of cycling genes using expression data the above papers relied on the regulatory information in each species. Such information may not be accurately available for all genes and transcription factors in various species. Specifically, the networks they relied on were not systems specific but rather general, and the goal was to extract global and local similarities as opposed to the cell cycle oriented goal in our paper.

A number of papers used belief propagation to combine different biological data sources. These include the physical networks model by Yeang *et al*. (2004) and methods for determining protein functions (Letovsky and Kasif, 2003). These are very different in their goal from our work, and use different types of data. In addition, these papers did not try to combine information from different species, as we do here.

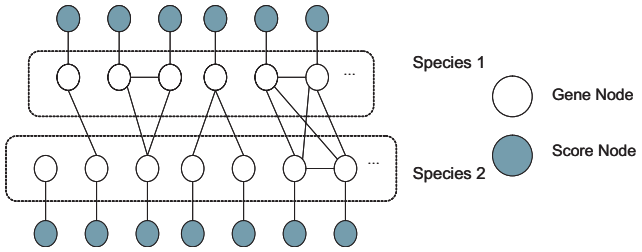


Fig. 1. A graphical model for two species. Dark nodes are score nodes, representing the score derived from such experiments. The lighter nodes are gene nodes. Gene nodes are connected by edges if their sequence is similar.

2 MODELING EVOLUTIONARY CONSERVATION USING GRAPHICAL MODELS

We formulate the problem of assigning cyclic status to genes using probabilistic graphical models. In such models, random variables are represented by nodes in a graph and conditional dependencies are represented by edges. The structure of the graph and the conditional dependencies it implies specify a joint probability distribution on the random variables. By taking advantage of the structural relations in the graph, efficient algorithms have been proposed to either learn the parameters of graphical models or do inferences on learned models.

Here we use Markov random fields (MRF) to represent dependencies between genes in different species. Unlike Bayesian networks, MRFs are undirected graphical models, in which dependency among nodes is represented using potential functions. There are two types of nodes in the graph we use for this problem (see Figure 1). The first represents genes and the second represents expression scores from the related cell cycle experiments. Edges between gene nodes correspond to sequence similarity, and carry a weight which depends on that similarity. These edges are used to capture the conditional dependencies of phylogenetically related genes. All edges between a gene node and its corresponding score node have the same weight and correspond to the gene nodes' potentials.

To generate the edges between potential homologous genes, we run BLAST between all pairs of genes in the two species. We insert an edge between two gene nodes (either belonging to the same species or to two different species) if their BLAST score is higher than a fixed threshold. We use a conservative cutoff such that we are fairly confident that when an edge is added to the graph, the two genes it connects are very likely to be homologous. While we use a cutoff to determine whether we place an edge or not, edges that are present in the graph are weighted based on their BLAST score. The resulting graph comprises of a set of connected components, as demonstrated in the diagram in Figure 1.

To represent the latent status of a gene (whether or not it is a cell cycle gene) we associate a hidden variable C_i with each gene node. $C_i = 1$ means that this gene is cell cycle regulated, otherwise $C_i = 0$.

Based on the definitions above, the joint probability distribution over the random variables C_i of this model is defined as follows (Pearl, 1988)

$$L = \frac{1}{Z} \prod_i \psi_i(C_i) \prod_{i,j} \psi_{ij}(C_i, C_j) \quad (1)$$

where ψ_i (C_i) is the node potential function (derived from the score node), ψ_{ij} (C_i, C_j) is the edge potential function, and Z is the partition function, i.e. the normalization term. Potential functions capture constraints on a single variable or between a pair of dependent variables. For example, if two gene nodes i and j are connected by an edge with a large weight, it is likely that they are functionally related. Thus, the potential function will penalize assignments that are different in the different nodes (e.g., setting C_i to 0 and C_j to 1). Below we discuss the potential function in detail.

2.1 Score distribution

A key to our algorithm is the derivation of an expression score which is consistent across all species used. Once such an expression score has been derived, each score node is assigned the corresponding gene's score, S_i. We assume that S_i is drawn from a mixture distribution. Specifically, we assume two different distributions (for each species): a cell cycle specific distribution, which applies to all genes that are cell cycle regulated, and a null, or background distribution which applies to all other genes.

An important practical issue is to choose the form of the two component distributions of the S_i scores. While the Gaussian distribution has been successfully applied to model expression values, here we are modeling scores that are derived from such values, and not the values themselves. In many cases, such scores are derived by taking the max value of some transformation. Cell cycle score calculation involves taking the maximum peak of the expression time series or the Fourier transform and the resulting distribution often has a heavy tail and is more appropriately modeled as an Extreme Value Distribution (EVD). This heavy tail property is clearly noticeable in the scores assigned to known cycling genes as can be seen in Figure 3.

The EVD is defined using two parameters: location (a) and scale (b). Its PDF is given by:

$$p(x) = \frac{1}{b} e^{-\exp\{\frac{a-x}{b}\}} e^{\frac{a-x}{b}}$$

The location and scale parameters of EVD are similar to the mean and variance parameters of the Gaussian distribution. As in a Gaussian, they control the mode and the spread of the distribution, though they do not necessarily correspond to the mean and variance. Using the EVD mixture model we need to fit four parameters for each species a_0, b_0, a_1, b_1 where

$$S_i \,|\, C_i = 0 \sim EVD(a_0, b_0)$$
$$S_i \,|\, C_i = 1 \sim EVD(a_1, b_1)$$

The values of these parameters are fitted to the score distributions using an EM-type algorithm. As with any EM algorithm, the initial guess plays an important role in reaching a good local maximum. To initialize the parameters for the null distribution we permute each of the original time series randomly to simulate the expression levels of non cell-cycle genes. Scores are calculated from these artificial expression data, and are subsequently used to estimate the parameters of the null score distribution. To initialize the score for cell-cycle genes, we compile a list of such genes that appear in the corresponding papers and use the scores of these genes to derive a maximum-likelihood estimate of the parameters.

2.2 Node potential function

The node potential function is defined using Bayes rule as

$$\psi_i(C_i) = Pr(C_i \,|\, S_i) = \frac{Pr(S_i \,|\, C_i) Pr(C_i)}{Pr(S_i \,|\, C_i = 0) Pr(C_i = 0) + Pr(S_i \,|\, C_i = 1) Pr(C_i = 1)}$$

Using the EVD mixture assumption, the potential function becomes

$$\psi_i(0) = Pr(C_i = 0 \,|\, S_i) = \frac{t_{i0}}{t_{i0} + t_{i1}},$$

$$\psi_i(1) = Pr(C_i = 1 \,|\, S_i) = \frac{t_{i1}}{t_{i0} + t_{i1}}$$

where

$$t_{i0} = (1 - P_c) \cdot \frac{1}{b_0} e^{-\exp\{\frac{a_0 - S_i}{b_0}\}} e^{\frac{a_0 - S_i}{b_0}}$$

$$t_{i1} = P_c \cdot \frac{1}{b_1} e^{-\exp\{\frac{a_1 - S_i}{b_1}\}} e^{\frac{a_1 - S_i}{b_1}}$$

and P_c is a prior probability for cycling genes in the species to which i belongs.

In practice, we require $b_0 = b_1$ so that the two score distributions have a similar spread. This guarantees that the posterior score will have the same ranking as the expression scores when there are no edges in the graph.

2.3 Edge potential functions

Our edge potential functions capture the a-priori functional similarity between gene pairs. This is based on our assumption regarding evolutionary conservation of gene functions, namely, that genes that are highly similar in sequence are likely to be similar in function. We use BLAST (Altschul *et al*., 1997) to determine sequence similarity. As mentioned earlier, we do not transform these BLAST scores into binary features. Rather, we use the similarity score to determine the edge potential which penalizes contradictory assignments. The penalty is proportional to how close the two genes' sequences are.

For each query sequence, the BLASTALL program returns an E-value and a bit score S. The relation between them is $E = mn2^{-S}$ where m is the length of the query sequence and n is the length of the genome of the second species. Note that bit scores are not ''symmetric'' as they depend on the total genome length. To overcome this, and generate a single similarity score for pairs of genes we set the weight on edge (i, j) to

$$w_{ij} = \frac{1}{2}(b_{ij} + b_{ji})$$

where b_{ij} is the BLAST bit score of gene i against gene j. Using $w_{i,j}$ we define the edge potential as

$$\psi_{ij}(C_i, C_j) = 2^{-\lambda w_{ij}(C_i - C_j)^2}.$$

This potential function penalizes assignments that do not agree between connected nodes. λ is an externally specified parameter that controls the impact of edge potentials relative to the node potentials.

3 LEARNING THE PARAMETERS OF OUR MODEL

The model parameters we need to learn are the score distribution parameters for each species. We learn the score distribution parameters (a_0, b_0, a_1, b_1) in an iterative manner using an EM-style algorithm. We start with an informative guess for the score parameters, as mentioned above. Based on the score distributions we determine a posterior assignment to nodes using belief propagation, as we discuss below. Following convergence of the belief propagation algorithm we use the (soft) label assignments to update the score distribution parameters. We then repeat these steps by performing belief propagation again based on the updated score distributions and so forth until both the label assignment and score distribution parameters do not change anymore.

3.1 Iterative step 1: inference by belief propagation

To infer the node status variables C_i, we need to compute the marginal posterior label distribution on each gene node. This posterior is hard to compute directly because of the intractable normalization term Z in Formula (1). Fortunately, for these types of graphical models, we can use a standard belief propagation algorithm for inference avoiding the direct calculation of the Z term (Pearl, 1988). Note that our graph is loopy and thus the belief propagation algorithm is not guaranteed to converge to a global maximum. Still, as was shown in Yedidia *et al*. (2003) in practice these algorithms achieve good results in loopy networks as well.

The belief propagation algorithm consists of two steps: 'Message passing', where each node sends its current belief to all its neighbors, and 'belief update', where nodes update their belief based on the messages received. In our case the messages depend on the node's expression score and the belief of genes that are similar in sequence. The algorithm is summarized below.

(1) 'Message passing'. The messages sent by node i to node j about its belief in an assignment of 1 to j is:

$$m_{i,j}(1) \leftarrow \sum_{k=0,1} \left((\psi_i(k)\psi_{ij}(k,1) \prod_{n \in N(i) \setminus j} m_{n,i}(k) \right)$$

Where $N(i)$ is the set of neighbors of node i in the graph. Intuitively, this message informs j about i's agreement with an assignment of 1 to j. In order to determine this, i takes into account its own belief (from its score node), the strength of the edge between i and j and the belief of i's neighbors about the right assignment to i. For the belief in a 0 assignment we simply replace every 1 with 0 in the above equation. Note that the weighting parameter λ is already incorporated into the edge potential function and so it is incorporated into the message as well.

(2) 'Belief update'. The belief of i in an assignment of 1 is computed by setting:

$$b_i(1) = (1/v)\psi_i(1) \prod_{j \in N(i)} m_{j,i}(1)$$

where v is a normalization constant to make beliefs sum to 1. As can be seen, i's belief depends on both its original score and the messages it received from its neighbors about what they 'believe' should be assigned to i.

Table 1. Algorithm for combining microarray expression data from multiple species

Input
1. For each gene, expression score S_i
2. Graph structure (edge weights)

Output:
For each gene its posterior cycling status, C_i

Initialization:
For each species compute estimates for a_0, a_1 and b using permutation analysis and original lists

Iterate until convergence:
1. Carry out Belief Propagation to determine a posterior C_i for each gene
2. Use the computed posterior to recompute the EVD parameters for the score distribution in each species

3.2 Iterative step 2: updating the score distribution

Using the belief computed in the inference step, we update the score distribution parameters. Our goal is to maximize the auxiliary function $Q(\Theta,\Theta^{(g)})$, which is defined as the expected log likelihood of the complete data over the observed scores given the parameters $\Theta^{(g)} = (a_0^{(g)}, a_1^{(g)}, b^{(g)})$ at the g'th iteration.

We were unable to find a reference for deriving update rules for the EVD mixture distribution. We have thus derived these ourselves. In general, to derive an update rule for this distribution we need to simplify the Q function and separate the parameters into two terms which can be maximized independently. If we require that $b_0 = b_1$, then for each species we have three parameters: two location parameters a_0 and a_1 and one scale parameter b. We can find the location parameters that maximize Q easily if we know b, but there is no close form solution for b. However, we can use numerical methods to solve for b. The final update rules for each species are as follows

$$a_l^{(g+1)} = \frac{1}{\beta} \log \frac{\sum_{i=1}^{N} P_{il}}{\sum_{i=1}^{N} e^{-\beta S_i} P_{il}}, \quad l = 0, 1$$

$$b^{(g+1)} = \frac{1}{\beta}$$

where N is the number of genes in that species, P_{il} represents $p(C_i = l \mid S_i, \Theta^g)$, $l = 0$, 1, and β is the root of the equation:

$$\frac{1}{\beta} = \frac{\sum_{l=\{0,1\}} \sum_{i=1}^{N} S_i P_{il}}{\sum_{l=\{0,1\}} \sum_{i=1}^{N} P_{il}} - \sum_{l=\{0,1\}} \left[\sum_{i=1}^{N} P_{il} \frac{\sum_{i=1}^{N} e^{-\beta S_i} S_i P_{il}}{\sum_{i=1}^{N} e^{-\beta S_i} P_{il}} \right] \Big/ \sum_{l=\{0,1\}} \sum_{i=1}^{N} P_{il} \tag{2}$$

Equation (2) can be solved using linear line search since the reasonable range of β is not large. Note that the Newton-Raphson method does not work here, because the solution is very close to the local extrema of the function. See Appendix for more details.

We can also extend our model to use the Generalized Extreme Value Distribution which in some cases gives better results. For details please refer to our supporting website (Lu *et al*., 2006).

Our algorithm is summarized in Table 1 above.

4 RESULTS

We tested our algorithm on simulated and real biological data. For the biological species we selected budding yeast and humans. While budding and fission yeast are closer from the evolutionary standpoint, there is less complementary information for the set of cycling genes in fission yeast. In contrast, many of the human cell cycle genes have been extensively studied leading to good annotation databases for these genes. This makes it easier to evaluate a new list of cycling human genes when compared with a list for cycling fission yeast genes. Another advantage of looking at human instead of fission yeast is that it indicates that even if the two species are relatively far, they can still benefit from a joint analysis of their cell cycle expression experiments.

4.1 Simulated data

To test our model using simulated data we first generated the graph structure from the two species as discussed before. We then generated labels (i.e. cycling or not) for nodes in the graph using a Gibbs sampler method that took into account previously assigned neighboring nodes when assigning labels to individual nodes. See the supporting website (Lu *et al*., 2006) for complete details on the label assignment.

After generating the labels we assigned scores to nodes. We used two (overlapping) score distributions, one for the nodes with $C_i = 1$ and the other for those with $C_i = 0$. In all experiments we used a fixed distribution for one species. However, each experiment used a different distribution for the second species. These distributions varied in their separability, ranging from highly separable to highly overlapping (see Figure 2). We have next hidden the node assignments, and used our algorithm to infer these assignments. We repeated this process 10 times for each set of score distributions.

Figure 2 presents the results of two of these experiments. As can be seen, by relying on the graph structure we were able to improve the recovery of the true label assignments when compared to label assignments that are based on a cutoff of the score alone. As the separation between the two distributions became smaller the difference between the two methods became more apparent. For the less separable distributions our algorithm performed much better than the score only method by relying more heavily on the distribution of the other species.

These results indicate that under the evolutionary assumptions we stated in the introduction, our algorithm can improve the assignment of cycling genes and correctly recover more such genes.

4.2 Cell cycle expression data

To date, cell cycle expression was measured in more than six species. As mentioned above, the two most studied species are budding yeast and humans. Both provide access to a number of different validation sets, and are thus useful for comparison of our algorithm and score based methods.

We downloaded expression data from the corresponding websites for the budding yeast (Spellman *et al*., 1998) and human (Whitfield *et al*., 2002) cell cycle papers. All protein sequences for genes in these species were downloaded from the NCBI ftp server (http://ftp.ncbi.nlm.nih.gov). We used `Blastall` (Altschul *et al*., 1997) to score all pairs of genes in both species.

For this data we tested our algorithm using an Intel Pentium 4 PC with single 2.40GHz CPU. It typically took less than 6 minutes to converge.

Expression Scores: As mentioned earlier, it is important to use the same method to derive scores for genes in different species. We derived such scores based on the observed expression values. As was recently noted for yeast by de Lichtenberg *et al*. (2005), scores that look at both amplitude of the expression value peak as well as the peak in the spectrum around the cell cycle period seem to provide the best results for identifying genes using expression data only. We thus applied a similar method to extract such scores for all genes in both species (see the supporting website Lu *et al*., (2006) for details). To validate this method we compared our results to the benchmark provided by de Lichtenberg *et al*., (2005) and determined that our results for budding yeast were comparable to the best method presented in their paper. The results below use this scoring method. However, using the Whitfield *et al*. (2002) scoring method did not change the results. See the supporting website (Lu *et al*., 2006) for more details.

Comparison sets: As far as we know, this is the first method to combine sequence and expression data for the task of identifying cycling genes. In order to compare our results to previous methods we use two different alternative lists. The first list is the list of cycling genes (in each species) based on the expression score alone. As mentioned in the introduction, this is the method used by previous approaches. We have also compared our results to a more naive method for combining expression and sequence. Unlike our probabilistic approach, this naive method first computes ranking independently for each species based on expression score alone. Next, we identify conserved genes in both species and compute a joint ranking based on the average ranking for the orthologs in each species. While we do not claim that this method is ideal, it can at least serve as a baseline for evaluating the more sophisticated algorithm we present in this paper.

Identifying cycling human genes: To test the success of our algorithm for the task of identifying cycling human genes we used the GO human annotations. Of the 7254 human genes in the dataset we used, 498 were annotated by GO as cycling. We first ranked human genes using expression scores and the naive method mentioned above. Next, we ranked them using the posterior score computed by our algorithm.

Figure 3 (left) presents the precision recall curve for GO annotated cycling genes for the top ranked 1000 human genes. Based on the analysis in the original paper (Whitfield *et al*., 2002), roughly 1000 genes are determined to be cycling, which is why we focus on the top 1000. As can be seen, all three methods perform substantially better than a random ordering (dashed-dotted curve). Comparing our method with a score based method we see that while at the very high expression score (bottom left) we do slightly worse, overall, and in particular for lower scores our algorithm provides results that are superior to score based methods. Specifically, for the top 1000 genes our algorithm was able to recover 23% more genes (135 vs. 110) when compared to both, the score only method and the naive method for combining sequence and expression data.

Note that while we relied on the GO list for this analysis, it is not complete. It is possible that there are many cycling genes which are not on that list. Thus, the recall rate is probably much higher

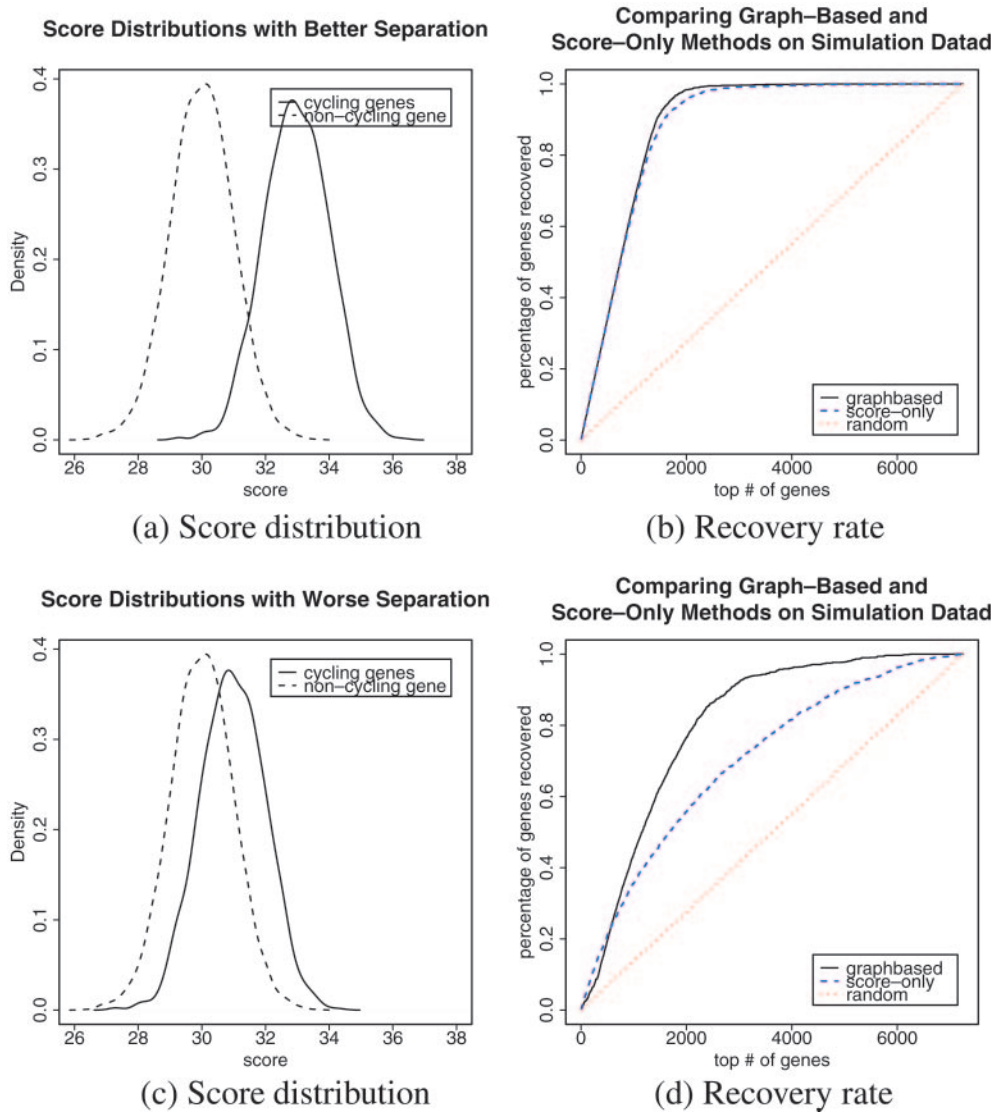


Fig. 2. Simulation results. 20% of the nodes were labeled with 1 and the rest were labeled with 0. (a) Score distribution and (b) Recovery rate for a well separated distribution. Both score based (dashed line) and graph based (solid line) methods were able to correctly recover the node assignments. (c) Score distribution and (d) Recovery rate for an overlapping score distribution. Note that while our graph based method can still achieve good precision and recall the score based method does significantly worse, especially for the higher recall rates (above 40%).

than the one we report here. Figure 3 (right) presents the expression score distribution of genes annotated as cycling in GO and genes that do not belong to this category. Note that there is substantial overlap between the two distributions making it hard for a score only method to identify a large set of cycling human genes. In contrast, our graph based method was able to partially overcome this problem by relying on the graph neighborhoods as we discuss below. Another issue is the possible homology bias of GO annotations. To account for this, we repeated the validation procedure using a smaller set of GO annotated human cell cycle genes. Specifically, we removed the 256 human genes that are annotated in GO as "cell cycle" based on homology evidence. Even with this reduced set of GO cycling genes our method outperforms the score based method by a similar margin. See website for details.

To further explore the differences between score based and graph based methods we examined the differences in cell cycle assignments between the two. We generated two lists. The first contained genes that appear in the top 1000 using our method but were not in the top 1000 of the score based method and the second contained genes in the top 1000 of the scoring method but not in our method. To test which of these list is more relevant we used GO to analyze both lists. On the supporting website (Lu *et al*., 2006) we present a number of figures comparing the GO enrichment p-values of both lists. As we show there, the majority of cell cycle related categories are more enriched for genes in the list derived based on our method when compared with the score based list.

Identifying cycling yeast genes We have used a dataset for protein-DNA binding (Lee *et al*., 2002) to compare our budding yeast results

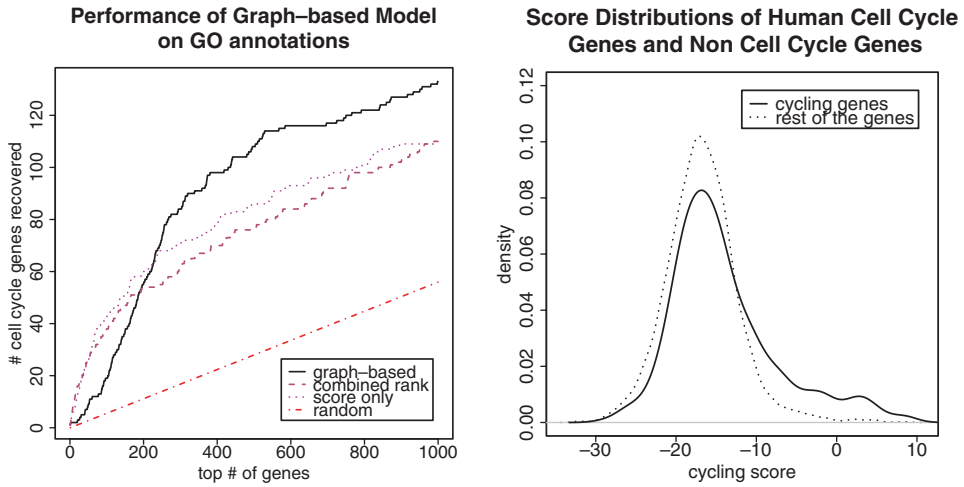


Fig. 3. Left: Identification of Human Cell Cycle Genes. The Y axis is the number of GO annotated human cell cycle genes in the top 1000 genes with highest posteriors. Our method (solid line) performs better than the score only method (dashed line) and the naive method for combining sequence and expression data (dotted line). Specifically for lower score thresholds our method achieves an improvement of over 20% over both other methods in terms of the number of accurately recovered cell cycle genes. Right: Score distribution for genes annotated as cycling in GO and the rest of the genes. As can be seen, these two distributions significantly overlap, making it hard to infer cyclic status from the expression score alone. The score distribution of the cell cycle genes has a heavy tail, and looks more like an Extreme Value Distribution than a normal distribution.

with the original list of Spellman *et al*. which was based on score alone. We extracted the binding information (p-value< 0.005 for the nine transcription factors that have been previously shown to play key roles in regulating cell cycle progression (Simon *et al*., 2001). We found 2.5% more interactions between these nine TFs and the top 800 genes on our list when compared with the Spellman list (621 vs. 606, note that a gene could be counted multiple times if more than one TF interacts with it). We also tested a stricter version of the binding data (p-value < 0.001). As with the higher p-value, our method still resulted in slightly more interactions (477 vs. 474) when compared to the score based list. While these improvements are far less dramatic than the results presented for the human data above, it still implies that our method can improve cell cycle assignment even for high quality datasets, like the yeast cell cycle expression data (Wichert *et al*., 2004).

Graph neighborhoods To further explore how our method helps in correct assignment of cell cycle status we have plotted two of the subgraphs in our graph. The shape of the nodes in each subgraph represents the species, and the different shades of node color represent the cycling expression score of the gene. Darker shades represent higher *expression* scores, and the darkest shade means that the expression score is within the top 1000 for human and top 800 for yeast. The first subgraph (Figure 4) contains members of the Rho family of genes in yeast and humans. These genes are involved in cell wall formation which is an integral part of the cell cycle system. Specifically, the cell wall integrity signaling pathway is controlled by Rho1 (Levin, 2005) . Based on its expression score Rho1 was not in the top 800 yeast genes. However, since its expression score is high enough, and since its local neighborhood is all assigned cyclic status, our algorithm assigns a posterior score that is at the top 800 for yeast genes allowing us to correctly recover this gene.

Why expression scores are not sufficient? Expression value, especially in time series experiments which usually do not contain repeats for individual time points, are very noisy. To determine why

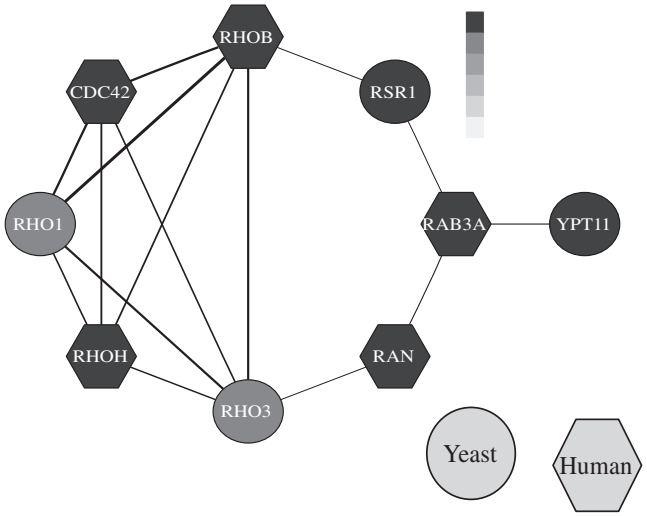


Fig. 4. Cluster containing the yeast cell wall gene Rho1. Node shades correspond to expression derived scores. Circles correspond to yeast genes and hexagons to human genes. Rho1 is not in the top 800 genes based on its score, but was identified by our algorithm because of its neighborhood.

our algorithm is able to correctly identify genes that cannot be detected using their expression score we looked at a number of genes that received high posterior scores and low expression scores.

One such gene is Mcm3, shown in Figure 5. Human Mcm3 is essential for the initiation of DNA replication and also participates in a checkpoint that ensures DNA replication is initiated once per cell cycle (Madine *et al*., 1995; Takei and Tsujimoto, 1998). On the top of Figure 5 we plot the graph neighborhood of Mcm3. As can be seen, it contains many known cycling genes from both species. On the bottom we plot the expression of Mcm3 in three different human cell cycle datasets (each done using a different

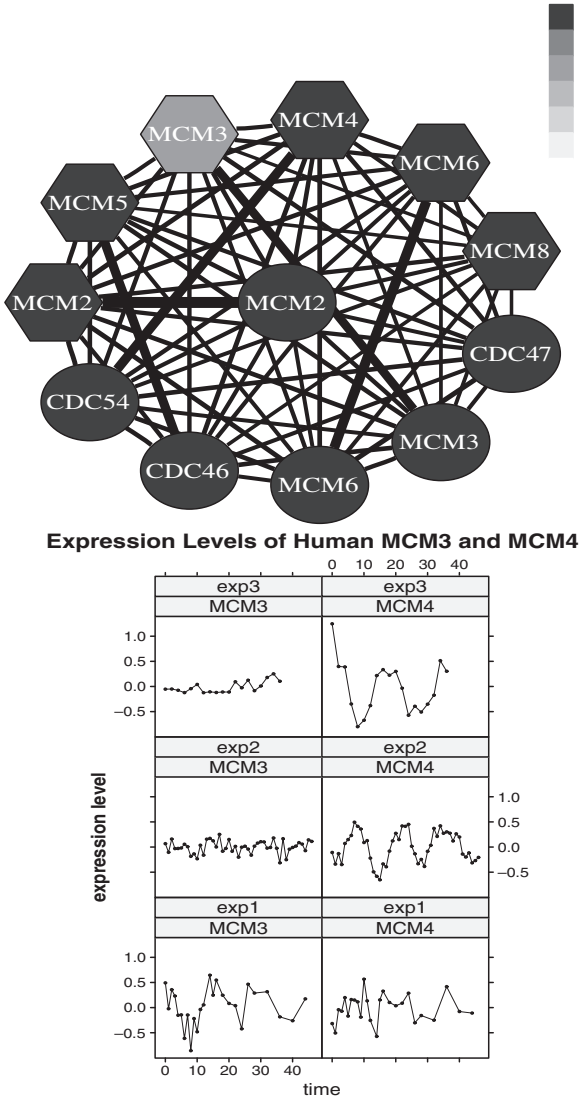


Fig. 5. Top: Gene cluster containing the human gene Mcm3, which is essential for the initiation of DNA replication. Bottom: plots showing the expression time series for both Mcm3 and Mcm4. Mcm4 scored in the top 15% but Mcm3 did not. Our algorithm was able to recover both genes.

arrest method). As can be seen, in at least one of these conditions Mcm3 seems to be cycling (bottom left). However, either because its expression levels are low in the other experiments or because of other experimental problems, it does not seem to be cycling in the other conditions. Using expression data alone, we would not assign a cyclic status to this gene. However, because of its medium expression score and its strong neighborhood score, our algorithm was able to correctly determine that it is a cycling human gene.

5 CONCLUSIONS AND FUTURE WORK

Many researchers have used gene expression experiments to study biological systems in various species. We presented an algorithm that combines information from studies in multiple species for the task of identifying cycling genes. Our algorithm constructs a graph where nodes represent genes and edges represent sequence similarity. We then use belief propagation to update the status of genes based on their graph neighborhood.

We applied our algorithm to combine cell cycle expression data from budding yeast and humans. Using our approach we were able to recover a more accurate set of cycling human genes when compared to the score based methods that have been used in the past. We have also shown that by looking at the neighborhood extracted from the graph we can infer properties that cannot be determined using expression alone.

While this paper focuses on cell cycle analysis, our algorithm is general and can works with any expression data as long as an expression score can be extracted from that data. An obvious future direction is to apply it to other biological systems that have been studied in multiple species such as immune response and circadian rhythm. Another direction is to combine regulatory data with the sequence data we currently use to infer sets of genes that are conserved in terms of regulation, sequence and function across multiple species.

ACKNOWLEDGEMENTS

This work was partially supported by NSF CAREER award 0448453 to ZBJ and by a tobacco settlement grant from the Pennsylvania department of health.

REFERENCES

Altschul,F.S., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, **25(17)**, 3389–3402.

Bar-Joseph,Z., Farkash,S., Gifford,D., Simon,I. and Rosenfeld,R. (2004) Deconvolving cell cycle expression data with complementary information. *Bioinformatics*, **20 Suppl 1**, I23–I30.

Bar-Joseph,Z. (2004) Analyzing time series gene expression data. *Bioinformatics*, **20(16)**, 2493–2503.

Bergmann,S., Ihmels,J. and Barkai,N. (2004) Similarities and differences in genome-wide expression data of six organisms. *PLoS Biol.*, **2(1)**, e9.

Bilmes,J.A. (1998) A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report TR-97-021, UC Berkeley.

de Lichtenberg,U., Jensen,L.J., Fausboll,A., Jensen,T.S., Bork,P. and Brunak,S. (2005) Comparison of computational methods for the identification of cell cycle-regulated genes. *Bioinformatics*, **21**, 1164–1171.

Laub,M., McAdams,H., Feldblyum,T., Fraser,C. and Shapiro,L. (2000) Global analysis of the genetic network controlling a bacterial cell cycle. *Science*, **290(5499)**, 2144–8.

Lee,T., Rinaldi,N., Robert,F., Odom,D. and Bar-Joseph,Z. *et al.* (2002) Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, **798**, 799–804.

Letovsky,S. and Kasif,S. (2003) Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, **19(Suppl 1)**, 197–204.

Levin,D. (2005) Cell wall integrity signaling in saccharomyces cerevisiae. *Microbiol Mol Biol Rev*, **69**, 262–91.

Lu,X., Zhang,W., Qin,Z., Kwast,K. and Liu,J. (2004) Statistical resynchronization and bayesian detection of periodically expressed genes. *Nucl. Acids. Res.*, **32**, 447–455.

Lu,Y., Rosenfeld,R. and Bar-Joseph,Z. (2006) Supporting website. http://www.cs.cmu.edu/∼lyongu/pub/cellcycle/.

Madine,M.A., Khoo,C.Y., Mills,A.D. and Laskey,R.A. (1995) Mcm3 complex required for cell cycle regulation of dna replication in vertebrate cells. *Nature*, **375**, 6530.

Menges,M., Hennig,L., Gruissem,W. and Murray,J. (2002) Cell cycle regulated gene expression in arabidopsis. *J Biol Chem.*, **277(44)**, 41987–42002.

Pearl,J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Rustici,G., Mata,J., Kivinen,K., Lio,P., Penkett,C., Burns,J., Hayles,G., Brazma,A., Nurse,P. and Bahler,J. (2004) Periodic gene expression program of the fission yeast cell cycle. *Nat. Genet.*, **36(8)**, 809–17.

Schliep,A., Schonhuth,A. and Steinhoff,C. (2003) Using hidden markov models to analyze gene expression time course data. *Bioinformatics*, **19**, i264–i272.

Sharan,R., Suthram,S., Kelley,R., Kuhn,T., McCuine,S., Uetz,P., Sittler,T., Karp,R. and Ideker,T. (2005) Conserved patterns of protein interaction in multiple species. *Proc Natl Acad Sci USA*. **102(6)**, 1974–9.

Shedden,K. and Cooper,S. (2002) Analysis of cell-cycle-specific gene expression in human cells as determined by microarrays and double-thymidine block synchronization. *PNAS*, **99(7)**, 4379–84.

Simon,I., Barnett,J., Hannett,N., Harbison,C., Rinaldi,N., Volkert,T., Wyrick,J., Zeitlinger,J., Gifford,D., Jaakkola,T. and Young,R. (2001) Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, **106**, 697–708.

Spellman,P.T., Sherlock,G., Zhang,M., Iyer,V., Anders,K., Eisen,M.B., Brown,P.O., Botstein,D. and Futcher,B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisia* by microarray hybridization. *Mol. Biol. of the Cell*, **9**, 3273–3297.

Stuart,J.M., Segal,E., Koller,D. and Kim,S.K. (2003) A gene-coexpression network for global discovery of conserved genetic modules. *Science*, **302(5643)**, 249–55.

Takei,Y. and Tsujimoto,G. (1998) Identification of a novel mcm3-associated protein that facilitates mcm3 nuclear localization. *J Biol Chem*, **273**, 22177–22180.

Whitfield,M., Sherlock,G., Saldanha,A., Murray,J., Ball,C. *et al.* (2002) Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Mol. Biol. Cell*, **13(6)**, 1977–2000.

Wichert,S., Fokianos,K. and Strimmer,K. (2004) Identifying periodically expressed transcripts in microarray time series data. *Bioinformatics*, **20**, 5–20.

Yeang,C., Ideker,T. and Jaakkola,T. (2004) Physical network models. *J Comput Biol*, **11(2–3)**, 243–62.

Yedidia,J.S., Freeman,W.T. and Weiss,Y. (2003) Understanding belief propagation and its generalizations. *Exploring Artificial Intelligence in the New Millennium*, pp. 236–239.

Zhao,L.P., Prentice,R. and Breeden,L. (2001) Statistical modeling of large microarray data sets to identify stimulus-response profiles. *PNAS*, **98**, 5631–5636.

6 APPENDIX

6.1 Derivation of update rules for EVD mixture model

Here are some facts of the Type I EVD, or the Gumbel distribution. The CDF and PDF of the EVD are

- CDF

$$F(x) = \exp\left\{-\exp\left[-\left(\frac{x-a}{b}\right)\right]\right\}, \quad -\infty < x < \infty$$

- PDF

$$p(x) = \frac{1}{b}\exp\left\{-\exp\left[-\left(\frac{x-a}{b}\right)\right]\right\}\exp\left\{-\left(\frac{x-a}{b}\right)\right\}$$

In the EM algorithm, we define the Q function to be $Q(\Theta, \Theta^{(i-1)}) = E[\log p(\mathcal{X}, \mathcal{Y} \mid \Theta) \mid X, \Theta^{(i-1)}]$ where $\mathcal{X}$ is the observed data, i.e. expression scores, and $\mathcal{Y}$ represents the hidden variables, i.e. the cycling status of the genes. In each E-step, we evaluate the above expectation, and in each M-step we maximize this expectation. For mixture models, the expectation can be written as (Bilmes, 1998)

$$Q(\Theta, \Theta^{(i-1)}) = \sum_{l=1}^{M}\sum_{i=1}^{N} \log(\alpha_l) p(l \mid x_i, \Theta^g) + \sum_{l=1}^{M}\sum_{i=1}^{N} \log(p_l(x_i \mid \theta_l)) p(l \mid x_i, \Theta^g)$$

where x_i is the i-th observed data point (i.e. the score S_i for the i-th gene), α_l is the mixing coefficient of the l-th component, p_l is the PDF for the l-th component, and $p(l \mid x_i, \Theta^g)$ from now on denoted as P_{il} for simplicity, is the probability x_i being generated by the l-th component, given the parameters Θ^g.

To maximize this expression, we can maximize the two terms independently. Using Lagrange multipliers, we solve for α_l that maximizes the first term, and get

$$\alpha_l = \frac{1}{N}\sum_{i=1}^{N} P_{il}$$

The maximization of the second term depends on the PDF of the component distributions. For the EVD mixture model, the second term becomes

$$\begin{aligned} B &= \sum_{l=1}^{M}\sum_{i=1}^{N} P_{il} \log(p_l(x_i \mid \theta_l)) \\ &= \sum_{l=1}^{M}\sum_{i=1}^{N} P_{il}\left[-\log b_l - \frac{x_i - a_l}{b_l} - \exp\left\{-\frac{x_i - a_l}{b_l}\right\}\right] \end{aligned}$$

Note that we also require the two components to have the same scale parameter, so we can drop the subscript and denote b_l as b. Now we maximize B by want to solve

$$\begin{aligned} \frac{\partial B}{\partial b} &= \sum_{l=1}^{M}\sum_{i=1}^{N}[-\beta + \beta^2(x_i - a_l) \\ &\quad - \beta^2(x_i - a_l)\exp\{-\beta(x_i - a_l)\}]P_{il} \\ &= -\beta\sum_{l=1}^{M}\sum_{i=1}^{N} P_{il} + \beta^2\sum_{l=1}^{M}\sum_{i=1}^{N} x_i P_{il} \\ &\quad - \beta^2\sum_{l=1}^{M} e^{\beta a_l}\sum_{i=1}^{N} e^{-\beta x_i} x_i P_{il} \\ &= 0 \end{aligned}$$

where $\beta = 1/b$. The above equation can be transformed to

$$\frac{1}{\beta} = \frac{\sum_{l=1}^{M}\sum_{i=1}^{N} x_i P_{il}}{\sum_{l=1}^{M}\sum_{i=1}^{N} P_{il}} - \sum_{l=1}^{M}\left[\sum_{i=1}^{N} P_{il}\frac{\sum_{i=1}^{N} e^{-\beta x_i} x_i P_{il}}{\sum_{i=1}^{N} e^{-\beta x_i} P_{il}}\right] \Big/ \sum_{l=1}^{M}\sum_{i=1}^{N} p(l \mid x_i, \Theta^g)$$

Define

$$f(\beta) = \frac{1}{\beta} - \frac{\sum_{l=1}^{M}\sum_{i=1}^{N} x_i P_{il}}{\sum_{l=1}^{M}\sum_{i=1}^{N} P_{il}} + \sum_{l=1}^{M}\left[\sum_{i=1}^{N} P_{il}\frac{\sum_{i=1}^{N} e^{-\beta x_i} x_i P_{il}}{\sum_{i=1}^{N} e^{-\beta x_i} P_{il}} e^{-\beta x_i} P_{il}\right] \Big/ \sum_{l=1}^{M}\sum_{i=1}^{N} P_{il}$$

and the β we are looking for is the root of $f(\beta) = 0$. In this case, since the root is near a local extremum ($\lim_{x \to +0} f(x) \to +\text{inf}$), the Newton-Raphson method can fail. Fortunately, we can simply use a root bracketing algorithm to search for it, because we don't expect the variances of the distribution to be too big.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e323–e331
doi:10.1093/bioinformatics/btl228

Quantification of transcription factor expression from Arabidopsis images

Daniel L. Mace[1,2], Ji-Young Lee[3], Richard W. Twigg[3,4], Juliette Colinas[3,4], Philip N. Benfey[1,3] and Uwe Ohler[1,*]

[1]Institute for Genome Science and Policy, Duke University, Durham NC, 27708, [2]Computational Biology and Bioinformatics Program, Duke University, Durham NC, 27708, [3]Department of Biology, Duke University, Durham NC, 27708 and [4]University Program in Genetics and Genomics, Duke University, Durham NC, 27708

ABSTRACT

Motivation: Confocal microscopy has long provided qualitative information for a variety of applications in molecular biology. Recent advances have led to extensive image datasets, which can now serve as new data sources to obtain quantitative gene expression information. In contrast to microarrays, which usually provide data for many genes at one time point, these image data provide us with expression information for only one gene, but with the advantage of high spatial and/or temporal resolution, which is often lostin microarray samples.

Results: We have developed a prototype for the automatic analysis of Arabidopsis confocal images, which show the expression of a single transcription factor by means of GFP reporter constructs. Using techniques from image registration, we are able to address inherent problems of non-rigid transformation and partial mapping, and obtain relative expression values for 13 different tissues in Arabidopsis roots. This provides quantitative information with high spatial resolution, which accurately represents the underlying expression values within the organism. We validate our approach on a data set of 122 images depicting expression patterns of 30 transcription factors, both in terms of registration accuracy, as well as correlation with cell-sorted microarray data. Approaches like this will be useful to lay the groundwork to reconstruct regulatory networks on the level of tissues or even individual cells.

Contact: uwe.ohler@duke.edu

Availability: Upon request from the authors.

Supplementary Data: http://tools.genome.duke.edu/generegulation/

1 INTRODUCTION

The development and spatial patterning of an organism is tightly controlled by differential gene expression in individual tissues and cells. Although a variety of factors contribute to this control of gene expression (e.g. microRNAs and epigenetic factors), one of the fundamental mechanisms is the binding of transcription factors (TF) to the promoter regions of genes, and the resulting networks of transcriptional control. While traditional biology has analyzed connections in these networks using a bottom-up approach (e.g. gene knockouts or knockdowns), technologies such as microarrays provide data for the inference of regulatory connections through the analysis of expression levels—often referred to as a top-down methods. However, the established way of measuring gene expression by DNA microarrays frequently averages over areas with different expression signatures and does not provide cues as to preferred spatial expression. To obtain a thorough understanding of gene regulation, we must move beyond these limits towards an accurate and detailed description of spatiotemporal (4-D) gene activity and regulatory interactions. High throughput digital microscopy has begun to deliver large datasets describing where a gene is expressed at a particular stage in living organisms. We are now faced with the task of how to use this rich information resource in combination with computational approaches with the aim of elucidating regulatory interactions in the development of multicellular organisms.

The process of extracting information from images is not new and has been particularly established for biomedical problems; examples include the mapping of brain scans and the automatic identification of breast cancer tumors (Maintz and Viergever, 1998; Woods *et al*., 1998). Recently, these techniques have begun to be adapted to molecular biology. In Drosophila, analysis of RNA in situ hybridization images has been used to identify expression patterns (Kumar *et al*., 2002; Peng and Myers, 2004). Due to the variability in staining, in-situ patterns are not capable of providing accurate expression values and are more of a qualitative nature.

Fluorescent proteins, such as Green Fluorescent Protein (GFP), can be used to quantitatively visualize the expression of a gene (Chudakov *et al*., 2005). It has been demonstrated that the intensity values from these fluorescent protein fusions are capable of recapitulating the underlying molecular biology of yeast with high confidence (Wu and Pollard, 2005). Quantification of fluorescent proteins in yeast does not need to address issues of attenuation due to depth or multiple tissue regions that are present in multicellular organisms. Additional work in sea urchins has shown that, by using a known injected fluorescent standard, one can correct for this attenuation and provide accurate measurements (Dmochowski *et al*., 2002). GFP reporter constructs have also been used to derive precise quantitative models of a small regulatory cascade in early Drosophila development (Jaeger *et al*., 2004). This work has

*To whom correspondence should be addressed.

demonstrated the potential for extracting expression profiles from confocal images.

Here we present an approach to automatically obtain transcription factor expression levels from GFP confocal images in Arabidopsis. In particular, we will consider longitudinal images of the root region. We have chosen the Arabidopsis root as a model because it provides a distinctive spatial patterning of cell differentiation, allowing us to restrict the current analysis to 2-D cross-sections (Benfey and Scheres, 2000). In addition, we also have a unique resource: tissue enriched microarray data at our disposition, which will provide a standard to validate our method (Birnbaum *et al*., 2003).

In order to identify and correctly map root tissues, we employ image registration algorithms. Image registration is a very broad subject with many applications to biological and biomedical data (Maintz and Viergever, 1998; Zitova and Flusser, 2003). After a brief overview of the data available to us (section 2), we describe the details of our registration process (section 3) to map an image onto a representative model—in our case, a labeled tissue map of a model Arabidopsis root. In section 4, we show that this method is capable of identifying and quantifying the expression profiles of 13 tissues in the Arabidopsis root, and we evaluate how well microarray and image-derived expression values correlate with each other. Section 5 addresses future developments and the implications of our results for the inference of regulatory mechanisms and pathways in multi-cellular organisms. An earlier version of this method was used in a large-scale study to assess the influence of post-transcriptional gene regulation on the expression of transcription factors (Lee *et al*., 2006). This work differs by utilizing new methods that expand our identification from 4 to 13 tissues, and also allow for analysis of images taken from all regions of the root.

2 DATA

2.1 GFP promoter fusion

Using the coding sequence for GFP, transcriptional fusion constructs were created by attaching the promoter region of the gene of interest (3kb upstream of the translation start site or the intergenic region—whichever is shorter) to the coding region of the GFP gene. In contrast to translational fusions—which incorporate the GFP as a domain into the protein—transcriptional fusions function as a marker for mRNA expression. The constructs were inserted into the genome with the assumption that its transcriptional regulation will be similar to that of the endogenous gene. While this concept ignores some of the regulatory steps of gene expression (e.g. translational/transcription inhibitions, chromatin modification, etc.), previous work has shown that, in Arabidopsis, this type of construct recapitulates tissue specific gene expression with high fidelity (Lee *et al*., 2006).

2.2 Images and image selection

The Arabidopsis images depict optical longitudinal sections of transcription factor GFP constructs taken from all three main zones of the root: meristematic (primary root growth and location of initials), elongation (elongation of cell size), and maturation (root hair growth) region (Figure 1). The use of longitudinal images allows us to identify 13 tissue regions (Figure 2): columella root cap, columella initials, cortex, cortex initials, endodermis, epidermis,

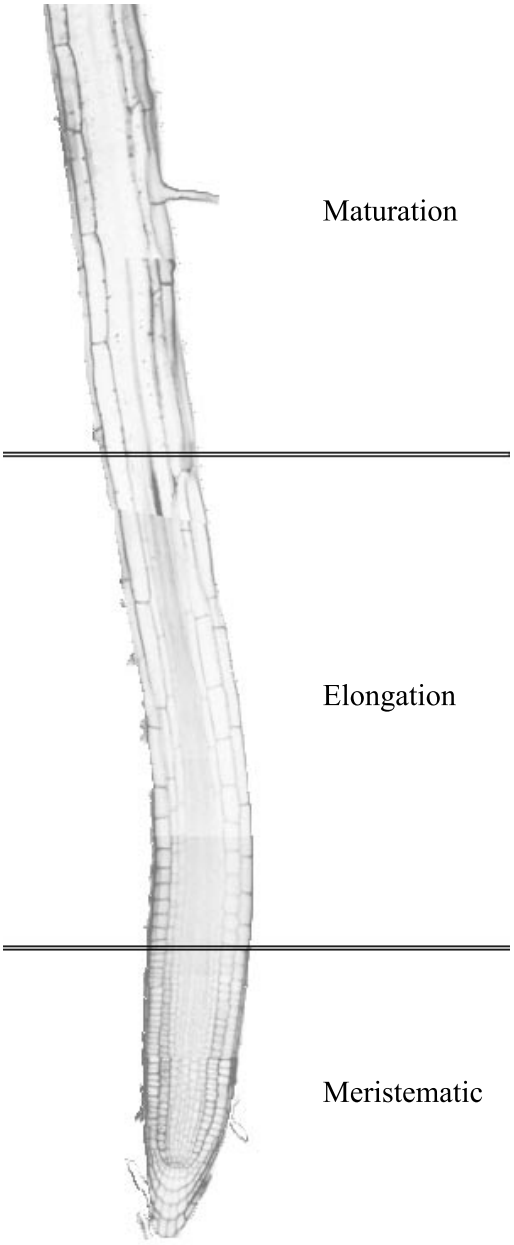


Fig. 1. Three main regions of the Arabidopsis root: meristematic, elongation, and maturation.

lateral root cap, lateral root cap initials, pericycle, pericycle initials, stele, quiescent center (QC), and vascular bundle (VB) initials. Tissues that cannot be distinguished in longitudinal images are the atrichoblast and trichoblast (epidermis), and the xylem and phloem (stele).

Images are composed of three channels: a red channel highlighting cell wall boundaries stained using a dye, a green channel containing the GFP expression, and a blank blue channel. Selection of images for comparison were based on the following criteria:

- The cell wall stain was strong on the external boundary and at least partially visible in the interior of the root.

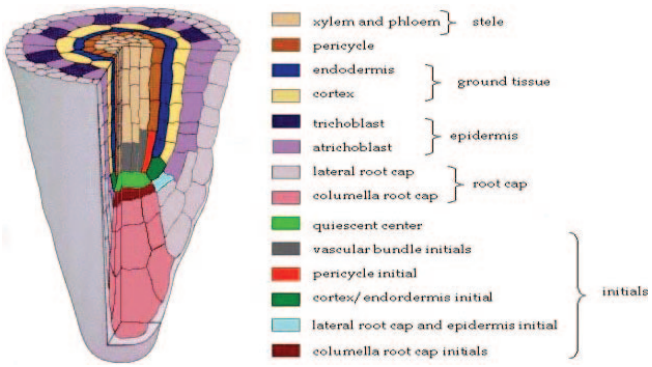


Fig. 2. A tissue map of the Arabidopsis root (Image taken from Benfey and Scheres (2000)).

- Roots were centered and not heavily skewed to one side of the image.
- Images were chosen from transgenic lines known to harbor detectable transcriptional fusions.

122 images representing 30 transcription factors met these criteria. 64 of them expressed in the elongation/maturation region, and the remaining 58 expressed in the meristematic region. To segment roots into the 13 tissue regions, we used an atlas image which contains a tissue map for a representative Arabidopsis root. We created this atlas by fusing two high resolution images: one of the meristematic zone up to the elongation zone, and one from the elongation zone to the maturation zone. Within this composite image, we marked the tissue regions as depicted by a standard template (Figure 2).

2.3 Fluorescent Activated Cell Sorting (FACS)

The tissue-specific microarray data is collected using a Fluorescent Activated Cell Sorting technique (Birnbaum *et al*., 2003). Arabidopsis roots with GFP expression enriched for a particular tissue are run through a fluorescent activated cell sorter. FACS separates cells expressing GFP from non-GFP expressing cells, obtaining the enrichment of cells from a individual tissue. RNA from the sorted cells is then analyzed on a microarray, providing tissue enriched gene expression data. Eight tissue regions were common to both image registration and the tissue enriched gene expression data: columella root cap, cortex, endodermis, epidermis, lateral root cap, pericycle, quiescent center (QC), and stele. We will refer to these expression values as T_{GFP} and T_{FACS}. The five tissues not present in the microarray data are the initials: columella initials, cortex/endodermis initials, lateral initials, pericycle initials, and vascular bundle initials. Due to lack of promoters specific to each type of initials, it is not currently possible to use FACS to isolate initials. Differences on what constitutes a specific tissue exist between the FACS and GFP data. First, the overall area that expression is averaged over differs: not all GFP lines used for FACS are expressed ubiquitously across all regions of the root, and the region the image is taken from might not encompass the full range that the FACS data is obtained from. Second, some GFP lines used for sorting have a partial inclusion of additional tissues that leads to a slightly convoluted FACS measurement for that tissue. Despite these minor differences (see supplementary data for full details), the regions are treated as homologous.

2.4 Results scoring metric

2.4.1 Registration scoring metric To determine the accuracy of the registration process, we modify a commonly used accuracy measure called Test Point Error (TPE) (Zitova and Flusser, 2003). TPE measures the accuracy of the registration process by creating a set of homologous points with the atlas image that are not used in the registration process itself, but are used as an accuracy measure for the registration process. Our modification to the TPE does not use fixed points themselves, but instead marked regions. These marked regions are a subset of the total cells in the image, manually chosen based on their clearly distinguishable cell boundaries by an expert. As a result, not all tissue regions may be marked due to difference in quality of staining and localization of images (8 tissues are specific to the meristematic region and are not present in elongation/maturation images). Our scoring method is formally defined as *total/matched* where

$$matched = \sum_{i=1}^{S} I(a_i = b_i)$$

allowing I to be the indicator function equaling 1 if a_i, b_i are equal and 0 otherwise and *total = size(s)* with $i \in s$ if $a_i, b_i \in [1, 13]$. The numbers 1–13 refer to a unique tissue, and 0 to no tissue mapping available (either a cell wall or a region outside the root).

2.4.2 FACS scoring metric The quality of GFP derived expression values will be assessed by comparison to respective FACS microarray data. By treating each data source as a random variable where each tissue is a sample (i.e. $X = T_{GFP}$ and $Y = T_{FACS}$), we can calculate the Pearson correlation value between the two data sets for every image. Comparing the correlations on the level of each image is required, as variations in gain, laser power, and pinhole settings between images (used to obtain maximum visual contrast) prohibit large scale correlation calculations. Images from the elongation and maturation zones did not contain GFP measurements for three of the tissue regions (QC, columella root cap, and lateral root cap), and both FACS and GFP measurements for these tissues were not included in the Pearson Correlation calculation.

3 METHODS

The system to quantify tissue-specific expression from images consists of three main steps. First, noise from the imaging process or normal root growth must be removed. Second, roots are registered to a master atlas image. Third, using the registered image and the atlas image, GFP levels are quantified, and tissue-specific expression values are obtained.

3.1 Noise removal and contour detection

Noise can result from the imaging process itself (such as blurring, the addition of speckle noise, etc.), or due to variability which naturally occurs in the root and which would present a difficulty during the later image registration stages. Morphological operators such as image closing, restoration, and thresholding are applied on all channels to eliminate the general noise present from imaging.

Two types of root variability exist that provide difficulties with registering the images: boundary cells and root hairs. Boundary cells are lateral root cap cells that have detached themselves from the rest of the root. This is a natural process as the boundary cells provide the lubrication needed for the root to penetrate the soil. Once removed from the tissue layer, these boundary cells often adhere to, or reside in close proximity to, the outer cell walls—making it difficult to accurately detect the shape of the actual root.

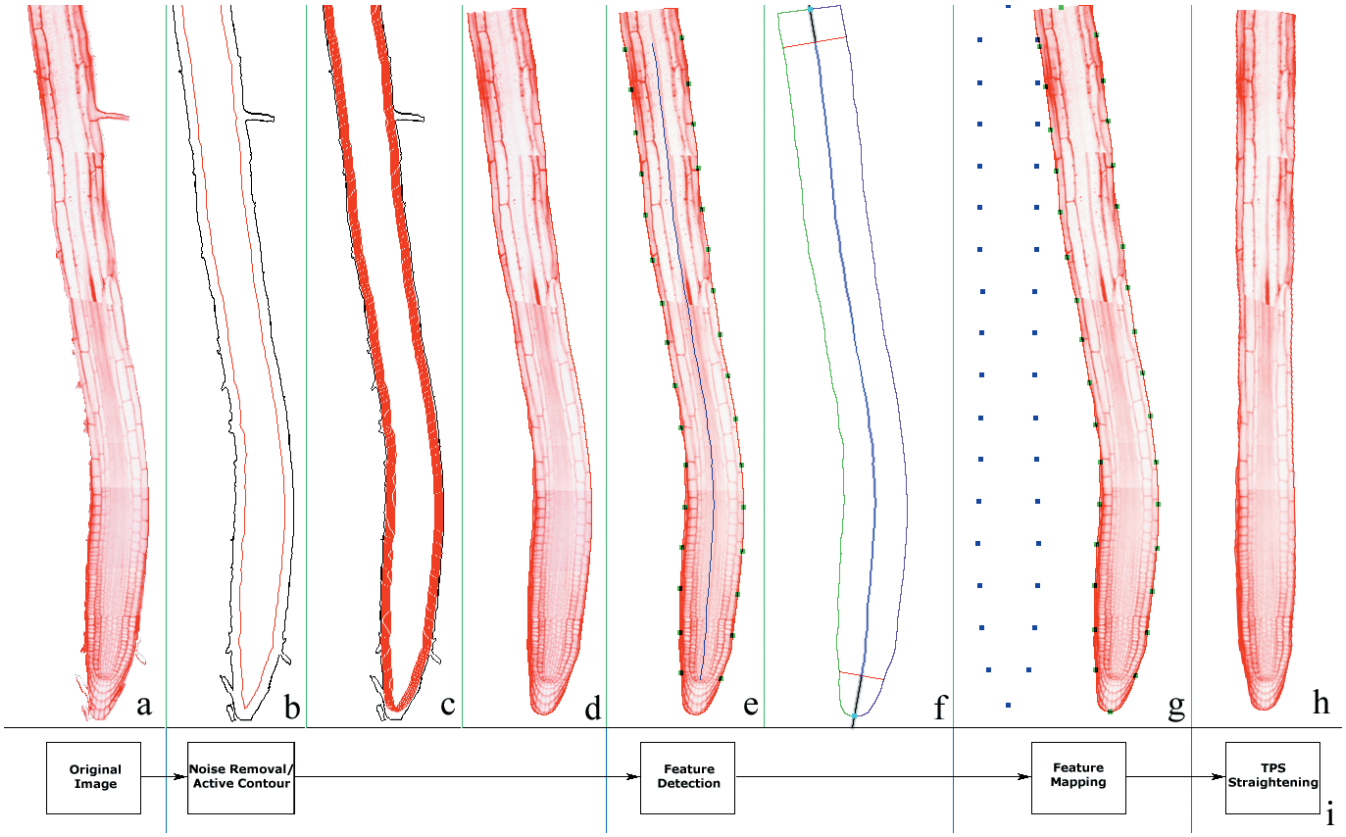


Fig. 3. Processing and straightening of the root. (a) Original root with root hairs and boundary cells. (b) Using the external noisy contour (black), we generate an internal eroded boundary (red) as a starting contour. (c) The snake algorithm is run until convergence, resulting in (d) a clean contour. (e) The medial axis (blue) is determined, and a subset of the cross-sectional cuts (green) are shown. (f) The medial axis (blue line) is extended (black line) at the two furthest end points (visually separated by the red line). The intersection of this extension with the outer contour provides us with the extreme points (teal points). Additionally, the resulting contour is divided into two sides (left and right- green and purple respectively). (g) The cross-sectional cut pairings, as well as the extreme points, (green) are used to create a straight set of points (blue). (h) Using these two sets of points, a Thin Plate Spline algorithm straightens the image. (i) Flow chart describing the general straightening process.

Unlike boundary cells (which have lost the connectivity and henceforth any cell signaling pathways), root hairs are a viable part of the root. Their occurrence along the longitudinal axis is not very predictable, which impedes any type of accurate registration process.

To remove these variabilities, we adapt a snake/active contour model. A snake is an iterative contour detection algorithms that can grow and shrink based on a set of force balancing equations (Kass *et al*., 1988). We use an improved active contour model called a Gradient Vector Field Snake, or GVF (Chenyang and Prince, 1998). Expanding on previous snake algorithms, a GVF snake is governed by two sets of forces: internal forces (such as elasticity/rigidness of the growing contour) and external forces (an external constant pressure force, viscosity, and a gradient vector field). With the exception of the gradient vector field, all parameters for these forces are user defined but kept constant for all images in the data set. A gradient vector field is a modification of a standard first order gradient edge map, in which the radius of the force field is increased. This causes it to extend its influence on the snake algorithm to areas outside those in close proximity to an edge.

For the gradient edge map, we use the external contour of the root. This contour is determined by performing a watershed segmentation (Luc and Pierre, 1991) on the image—segmenting the individual cells and background into different regions. For our current image set, we can safely assume that the region with the largest area can be labeled as background. In addition, regions with an area greater than one sixth and mean red and green intensity less than twice of the largest one are also labeled as background, to take cases into account in which the root partitions the background into two or more regions. Regions not fitting this criterion are considered part of the root. This creates a contour that contains all the root hairs/boundary cells which we seek to remove (Figure 3b, black outline). To initialize the starting contour for the snake algorithm, we perform a morphological erosion on the filled object. Due to differences in magnification/image size, this erosion is performed using a disk structuring element with variable size, which is set to 0.35 times the diameter of the root. This internal contour still contains some of the noise from the root hairs/boundary cells, but provides a smoother initial contour (Figure 3b, red outline). The algorithm is then run for a number of iterations adjusted to the size of root diameter (Figure 3c). By tuning the parameters of the snake algorithm (in particular, the elasticity/rigidness) once for our application, the final active contour can be adjusted so that it minimizes the amount of noise from boundary cells/root hairs. The resulting image is a clean smoothed external contour that does not contain root hairs/boundary cells (Figure 3d).

3.2 Registration

After removal of general noise and standardization with respect to boundary cells and root hairs, we can proceed with registering the images

to an atlas image. Two major issues must be considered for this registration process. First, the roots grow in a curved fashion, bending and twisting in response to the environment that they are growing in. This poses a non-rigid registration problem. Second, the images in the data set show different regions of the root, corresponding to a partial mapping problem. Non-rigid registration requires the calculation of complex transformation fields and cannot be solved by simple affine movements (e.g. scale, rotation, translation, etc.). Partial mapping requires the registration of an image under missing information—i.e. with occlusion of the object, or images showing only parts of a complete object (which is the case here). These two problems of non-rigid transformation and partial mapping are often mutually exclusive, and as such, we deal with them separately.

3.2.1 Non-rigid registration To allow roots to be aligned to a master atlas image, they need to be straightened. In order to do this, we will use a non-rigid transformation algorithm called Thin Plate Splines (TPS) (Bookstein, 1989). TPS is a transformation function which is derived from the physical bending energy of thin plates. TPS require a set of homologous points between a standard image and a reference image. A deformation field can be created based on the distance between the homologous points. This deformation field is then applied to every pixel in the entire image.

Successful straightening and registration using TPS is highly dependent on the set of homologous points that one chooses. Determining which, and how many, points to use for this mapping is referred to as feature detection. Successful feature detection for image registration requires that features are easily identifiable and abundant. This is often a problem with biomedical data, as the images do not contain features that fit these criteria (Zitova and Flusser, 2003). In our case, however, after the removal of boundary cells and root hairs, the contour of the root provides a source of features that fit both of these criteria.

Given the situation of partial mapping, as is the case for the root images, we encounter the problem of how to define a set of homologous points for the TPS. We address this by choosing a set of unique points for every image. This total feature set is then used to automatically derive a new set of points describing a straight root. Our feature set will contain two groups of features: a modification of the major axis endpoints (which we will refer to herein as the extreme points), and pairs of cross-sectional cuts, which are defined as the locations on the contour that result from the orthogonal bisection of the medial axis of the root.

The medial axis transformation (MAT) algorithm can provide us with the knowledge to derive these features (Ogniewicz and Ilg, 1992). The MAT function determines the medial axis by calculating the distance between every point in the interior of the object to the contour elements. The minimum distance of an internal point to its closest contour element is defined as the Voronoi distance. When the set of Voronoi elements is greater than one, i.e. the shortest distance to the contour is shared by two or more contour elements, it is considered part of the medial axis. To adapt for the partial occlusion of the images, the end points of the external contour are determined and the image is extended to create a new image three times the size of the original image, with the original being centered in the middle. Starting at the end points of the contour, we extend it into the added regions. Unlike traditional replicate padding of images, the extensions of the contour provide a better estimate of our expected shape of the root, and henceforth, a more accurate MAT estimate. While our initial noise removal algorithm is efficient in removing contour noise, the MAT is very sensitive to perturbations in the boundary of the object (e.g. natural distortions, small noise), and an additional step of pruning the MAT is required for removing any small branches. The extended regions are then removed.

Using the MAT, we can calculate a set of cross-section pairs. The MAT is treated as a continuous contour, and its curvature angle is calculated by using a standard first order derivative. Orthogonal lines are then drawn from the MAT, intersecting with the smoothed contour of the root. The intersection occurs on both sides of the root providing us with a pair of points where each point consists of an x and y component denoted as $[x_i, y_i]$. We will refer to this as a cross section pair: $p_i = \{l_i, r_i\}$, where l_i, r_i are the points on the left and right side of the contour respectively and $P = \{p_1, \ldots, p_n\}$ is the set of all cross section pairs (Figure 3e). The curvature of the root contour, in combination with the partial imaging of the root, leads to a subset of the cross-sectional pairs being incomplete as one of the pairs is occluded by the imaging process. To eliminate this abnormality, the location where pairs become occluded is determined for both ends of the roots. All regions of the root beyond this threshold are removed, resulting in an adjusted contour with blunt cut ends for those sides where occlusion was present.

This process leads to the set of cross-sectional cut points which we subsequently use to determine the remaining features—the extreme points of the root. Most major-axis algorithms for determining extreme points of objects are not appropriate here due to both the nonrigidness as well as the partial occlusion of the roots. The medial axis is trimmed to 70% of its normal size to eliminate small perturbations occurring at the ends. Using both end points of the trimmed medial axis, the extreme points are extrapolated to intersect with the new adjusted contour of the root labeled as $E = \{e_t, e_b\}$—the top and bottom end points respectively. These intersection points represent the extreme points of the root, and by definition, partition the root into two separate sides (Figure 3f).

Given this feature set selected from a given image, we can proceed to create a homologous mapping to an approximately straight root. The original set of points can be separated into two groups: the extreme points of the image $E = \{e_t, e_b\}$ (teal points Figure 3f), and the pairs of cross-section pairs $P = \{p_1, \ldots, p_n\}$ (green dots Figure 3e). We define a new set of points M based on their location along the root, as the ordered set of the middle point between each cross-section pair $c_i = [\frac{1}{2}(l_{i_x} + r_{i_x}), \frac{1}{2}(l_{i_y} + r_{i_y})]$, and the two extreme points defined above, leading to $m \in M = \{e_t, c_1, \ldots, c_n, e_b\}$. In our first step in deriving a new set of points, we define two distance functions D_1 and D_2. The first function, D_1, determines the distance between the ordered middle points and the first extreme point $D_1(i) = |m_i - e_t|$. The second distance function, D_2, measures the distance between the middle points and their respective cross-sectional pairs: $D_2(i) = \frac{1}{2}|l_i - m_i| + \frac{1}{2}|r_i - m_i|$.

Using these middle points and distance functions, we can now define a new set of straightened points E^*, P^*, M^*. We additionally use a parameter a_x, defined as the medial x-axis location of the image. Starting from the point furthest away from the tip of the root, we set the x-coordinates of our homologous middle points to this medial axis $m^*_{ix} = a_x \forall m^* \in M^*$. The y-coordinates for these middle points are determined from the first distance function $m^*_{iy} = D_1(i)$. This maps the new middle points along the medial x-axis, separating them by the same distance between their original middle points. The location of each pair of cross-sectional points is then determined by translating each point in the pair by an equal distance in opposite directions from the medial x-axis; $l^*_{ix} = a_x - D_2(i)$ and $r^*_{ix} = a_x + D_2(i)$, and the y-coordinate is set to its middle point $l^*_{iy} = r^*_{iy} = m^*_{iy}$, creating a new set of pairs $P^* = \{p^*_1, \ldots, p^*_n\}$. The resulting set of points $F^* = \{e^*_t, p^*_1, \ldots, p^*_n, e^*_b\}$ provides a mapping from our original image $F = \{e_t, p_1, \ldots, p_n, e_b\}$ onto a straight root (Figure 3g).

This homologous mapping of points finally provides us with the information needed to perform the TPS transformation. A transformation field is calculated from these sets of points and is applied to every pixel in the image. Due to the complexity and memory requirements of the TPS function, the two extreme points and a subset of the cross-sectional pairs (15 equidistantly spaced pairs) are used. The resulting image has eliminated most of the curvature and non-rigid abnormalities that exist on the contour of the root (Figure 3).

3.2.2 Partial mapping The final registration process addresses an affine registration between two images: the straightened root we have just obtained from the TPS registration, and the master atlas image which provides the tissue label information. An affine registration consists of minimizing a scoring metric. Traditional affine registration parameters encompasses rotation, skew, scale (both in x and y), and translation (both in x and y). In the process of root straightening, the TPS has already restricted the transformations required to register the root. Fixed along the center of the image with the root tip pointing to the bottom of the image, rotation, skew,

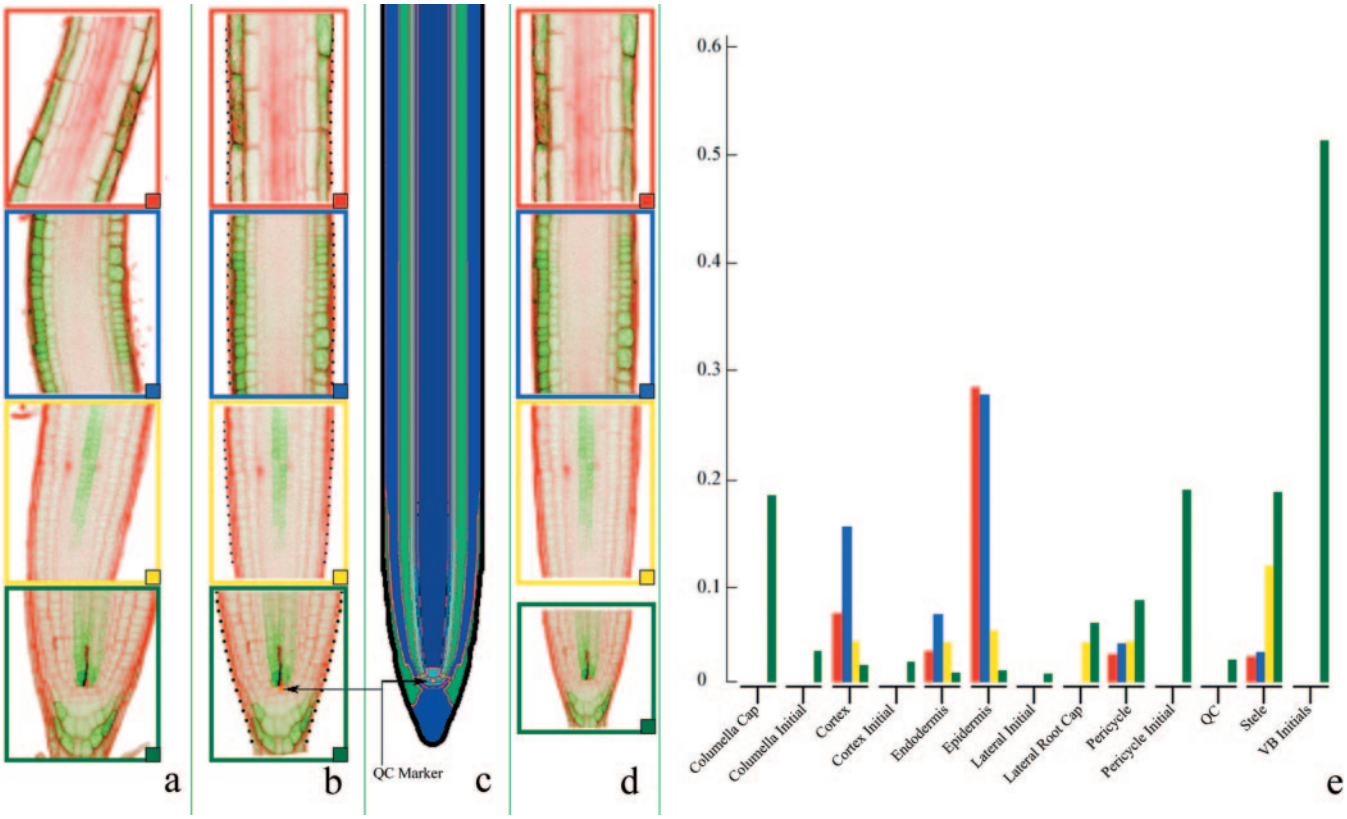


Fig. 4. Summary of the registration process. (a) A series of images (red, green, blue, yellow frames) with boundary noise and distortions is processed by the non-rigid registration algorithm, resulting in (b) straight roots. Points sampled on the boundary element (black) as well as an artificial QC marker (orange) where appropriate are used. (c) A master atlas image is used to describe the tissue region mapping of the root with superimposed contour (black) and QC marker (orange). (d) Minimizing the Hausdorff distance between the two sets of points, the scale and translation are determined using an iterative parameter optimizer. (e) Expression values are determined for each image (red, green, blue yellow respective to frame color) by summing the intensity values from the green channel in (d) and binning them by their tissue type as determined in (c). The expression values are normalized by dividing them by the total number of pixels per tissue type and range between 0 and 1.

and translation in x have already been determined. We assume that the scale is the same in both the x and y coordinates and will be treated as one parameter. The remaining degrees of freedom are then the translation along the y-coordinate and the scale of the image.

The affine scoring metric is motivated by the Hausdorff score for partial mapping (Huttenlocher *et al*., 1993). The Hausdorff scoring metric is formally defined as:

$$H(A,B) = max(h(A,B), h(B,A))$$

where

$$h(A,B) = \max_{a \in A} \min_{b \in B} (\mid a - b \mid)$$

with $\mid a - b \mid$ being the distance between a and b and $A = \{a_1, \ldots, a_p\}$ and $B = \{b_1, \ldots, b_q\}$ are two sets of points. Here, we modify this Hausdorff scoring metric to be

$$H(A,B) = \sum_{i}^{n} h(A_i, B_i)$$

where i denotes a subgroup of points and A and B are points in our image and atlas respectively. The first group is the subset of points we used for the TPS straightening, i.e. the contour of the root $A_1 = P^*$. For images taken from the meristematic region, the second group is a single point denoting the center of QC that is placed by an expert after the TPS process ($A_2 = Q^* = \{q^*\}$). This placement of a marker is currently necessary, as the internal cell staining is not robust enough to automatically determine its location. This leads to the full set of points: $A = \{A_1, A_2\} = \{p_i^*, \ldots, p_n^*, q^*\}$. Identical contour and QC markers are pre-determined and marked in the atlas image $B = \{B_1, B_2\} = \{p_i, \ldots, p_n, q\}$. A number of numerical optimization algorithms are appropriate; here we use a Particle Swarm Optimizer (PSO) (Kennedy and Eberhart, 1995). We limit the range of scale values from 0.1 to 10, and translation values from 0 to 2500. The optimization converges in less than 400 iterations (Figure 4d).

3.3 Quantification

We now have two images with identical dimensions, from which we proceed to extract expression values. The first image is the expert created atlas image describing the tissue map of the root (Figure 4c). The second is the result of the affine registration, with the green channel detailing our gene expression values (4d). For every pixel in the registered image, the intensity value is summed and binned according to the tissue map as defined by the homologous point in the atlas image. The thirteen tissue expression values are normalized by dividing the total expression values by the total area that each tissue region occupies (4e). This helps to normalize against the occlusion of certain tissue layers due to imaging and provides standardization similar to microarray data.

3.4 Image processing/data analysis

Most of the image analysis was carried out using the Matlab Image Processing Toolbox (IPT), with the exception of the Thin Plate Splines

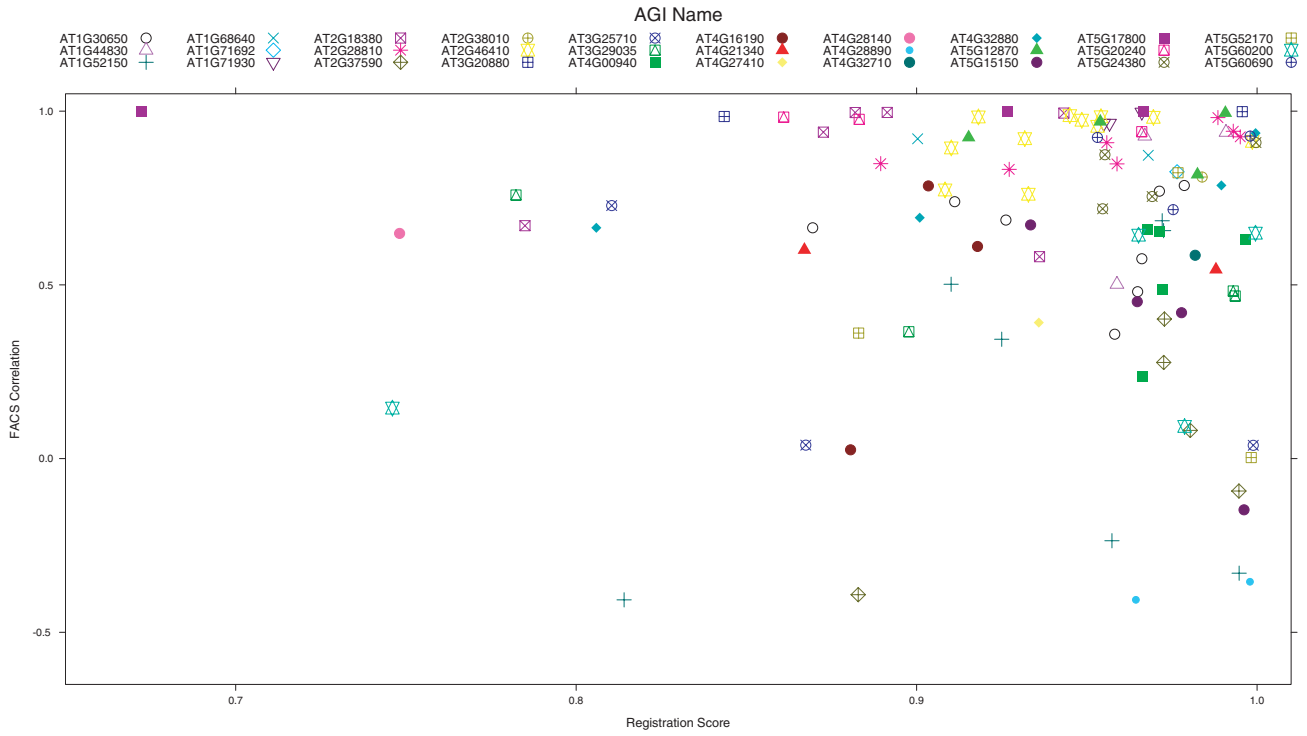


Fig. 5. Comparison of image registration (x-axis) and FACS correlation (y-axis) scores. Arabidopsis Gene Index (AGI) identifiers are given. The figure shows registration scores between 0.65 and 1.0 and correlation scores between −0.6 and 1.0. No other data points fell outside this region.

algorithm (Dollar, 2006) as well as the GVF Snake algorithm. The Hausdorff partial mapping and Particle Swarm Optimization are implemented in C#. Statistical analysis was performed using the R Statistical Language (R Development Core Team, 2005).

4 DISCUSSION

We applied this prototype to a data set of 122 GFP images depicting the expression of 30 transcription factors in different regions of the root. For 7 out of the total data set of 122 images, the system was unable to eliminate noise and perform the straightening. This was due in most part to boundary cells/root hairs being present at the edges of the images—a known limitation of our noise removal algorithm. For images requiring QC marking, we were unable to unambiguously locate the QC region in 5 of the images. The remaining 110 images were passed on to the second phase of registration and quantification.

Figure 5 shows the scatterplot of registration score on the x-axis and FACS correlation score on the y-axis. The majority of the images were successfully registered to the master atlas image: The mean registration score was 0.93, i.e. only 7% of the root is mapped to the wrong tissue type. The FACS correlation scores had a mean of 0.64. Considering that with exception of the QC marker, no root feature was manually marked to help register the images, the high accuracy of the registration process is very encouraging. It is notable that low registration scores do not necessarily lead to bad FACS correlation values: a mis-registration of a tissue layers may occur in a location where there is no GFP expression and have no effect on the correlation score.

While the FACS correlation scores had an overall good average of 0.64, a portion of these were rather poor. It is apparent from Figure 5 that several of these lowly correlated values are clustered within image groups of the same gene or line, suggesting potential issues with the promoter fusion of the GFP reporter constructs, or with the probe for the FACS data. The mean correlation score of 0.64 is reminiscent of previous studies for expression analysis, where it was found that correlations between platforms varied from 0.46 to 0.83 (Kim, 2003; Park *et al*., 2004; van Ruissen *et al*., 2005).

Lowly expressed genes. Poor correlation scores between platforms are frequently contributable to various sources of noise in different expression analysis platforms, and are increasingly observed for low expression values. Limiting our correlation calculation to FACS data where the median tissue expression is greater than 150 (used as noise threshold for Arabidopsis microarray data (Lee *et al*., 2006)) we increase our mean correlation to 0.70, thus reaffirming that lowly expressed values are more likely to have a negative effect on our correlation scores. In addition to the standard noise conditions present in microarray experiments (e.g. hybridization, background fluorescence, probe ambiguity etc.), a potential source arises from the FACS sorting of the data. While traditional microarray experiments use one sample per experiment, our data set requires 8 different sorting and microarray experiments, increasing the likelihood for biological variability.

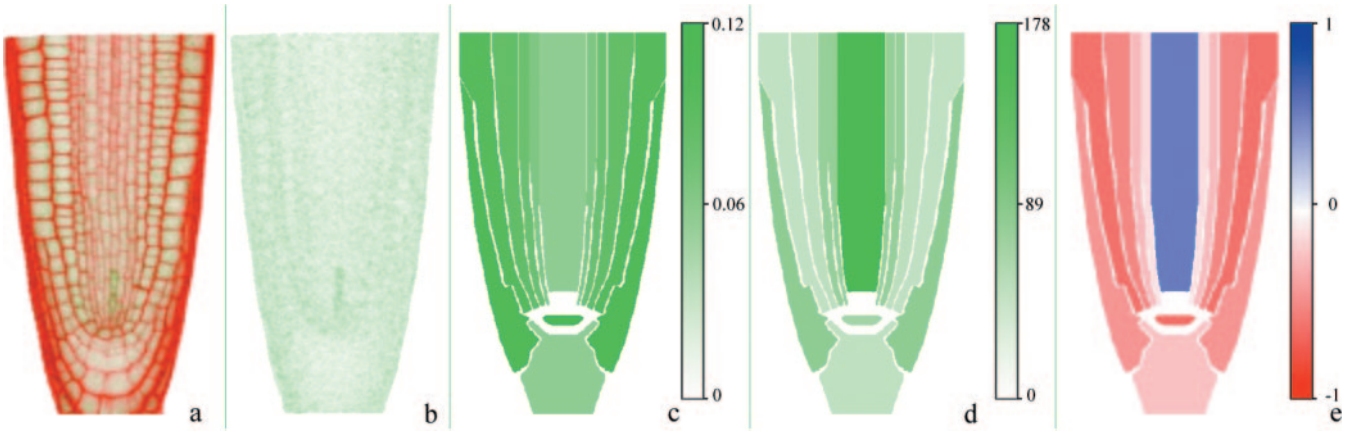


Fig. 6. (a) Evaluation of an image with low correlation value (ATG2G37590). (b) The filtered green channel with actual intensity values. (c) The scaled expression values of the GFP quantification, contrast enhanced to maximum intensity values (0.12). (d) The scaled FACS expression data contrast enhanced to maximum FACS expression (178). (e) Differences between scaled expression. Blue regions denote tissues where the FACS data is higher than the GFP data, while red regions denote GFP data being higher than FACS data.

Part of the noise in the FACS comparison originates from our approach to quantifying expression levels. Autofluorescence in the plant results in a mean background noise of 5–20% of the maximum possible intensity value in the green channel. On examining the images with the lowest correlation values, we noticed that a majority of these were from images with GFP expression levels barely above this autofluorescence level, suggesting that the correlation values were skewed by the background in the green channel. In addition to this background noise, our method did not take into account the attenuation due to depth which affects the inner tissues of the root, such as the endodermis, pericycle and stele. An initial simple approach to normalizing these regions by multiplying the tissue types by 1.3, 1.4 and 1.5 respectively increased our mean correlation score from 0.64 to 0.7. A comprehensive study of this problem may therefore lead to a more systematic correction of GFP derived expression values.

Taken together, this suggests that improvements in increasing the signal to noise ratio in the green channel are paramount for the GFP quantification of lowly expressed genes.

Tissue-specific expression differences. The remainder of the lowly correlated genes suggests some inconsistencies in our images as well as in our approach in normalization of the data. Failure to differentiate between expression in the pericycle and endodermis can lead to low correlation values between the data sets. This occurs when large pinhole settings during the imaging lead to longitudal images showing expression in both tissue regions, but where radial images show expression in only one. This problem tends to occur in a subset of the images for a given gene. Visual inspection confirmed that low correlation values for some genes expressed above background were not caused by problems with the image analysis, but by actual differences in expression values as reported by microarrays and GFP. Such differences can originate from reporter constructs which do not fully recapitulate the expression of the native gene, or due to discrepancies in the tissue-specific expression data. In either case, our system can serve as helpful resource to point out and quantify such problems.

Cells versus tissues. Finally, the current normalization of expression by total tissue area can blur expression which only occurs in a subset of the tissue. A representative example was the gene AT2G37590, where GFP expression resided in a subset of the stele region as compared to uniform expression across the whole tissue (Figure 6). When its expression was averaged over the whole stele, it barely exceeded the background noise level. It should be noted that microarray data is prone to the same issue—in the example, it showed expression in the stele at a value of 179, again barely above the background threshold of 150.

5 SUMMARY/OUTLOOK

In this paper, we have presented a system for the automated quantification of gene expression levels from digital images of GFP reporter constructs. As a proof of concept, we successfully performed an automated registration of Arabidopsis roots, derived tissue-specific expression values of transcription factors, and demonstrated that these values correlate well with microarray expression data. The data set used for this evaluation was only of modest size. However, the number of images in gene expression databases of other model organisms organisms (Tomancak *et al.*, 2002) is easily on the order of tens of thousands, which demonstrates the growing need to adapt image analysis to problems in computational biology. In addition to the biological significance of our methodology, we have presented a unique approach to both a partial mapping as well as a non-rigid registration problem. The combination of these two problems often requires one to manually annotate images prior to registration.

Developing a universal method for image registration across all types of images is considered an intractable problem (Zitova and Flusser, 2003). Image registration often utilizes domain specific information—incorporating unique modifications in the image registration process to adapt for differences that are inherit to a specific set of images. In our model, we have adapted methods in noise removal, feature detection, and feature mapping that are specific to the elongated, symmetric shape of Arabidopsis roots. However, it is expected that the series of algorithms used in our approach will be

useful for other confocal images, particularly for approximately symmetric objects.

Many of the difficulties with quantification of GFP (attenuation to depth, large pinhole settings) can be addressed by expanding our work to 3D. Efforts are under way to scale up microscopy and imaging from one 2-D cross-section to a stack of images. All image processing algorithms used in our system were chosen because they offer adaptations to 3D image processing. In addition to the increase in precision for our quantification, the expansion to 3D will also allow us to differentiate our stele measurements into xylem and phloem, as well as our epidermis measurements into atrichoblast and trichoblast.

We note that our current method requires a step of manually marking the QC region in images taken from the meristematic region. Improvements in cell wall staining will likely allow for the automatic detection of this region using image segmentation algorithms, such as the watershed algorithm, to identify individual cells. Current attempts to automatically identify this region are not robust enough given the present staining technology. This is exemplified by the fact that several of the QC regions could not even be manually annotated by an expert and were subsequently removed from the analysis. An alternative here is to use a second fluorescent marker which is constitutively expressed in the QC cells. Adequately registering images on cellular resolution will also allow us to identify differential expression of genes within tissues (cf. Figure 4 (yellow and green framed images) and Figure 6).

Our system was evaluated using the 8 tissues common between the image and microarray data sets. In total, our image analysis identifies 13 unique tissues. As mentioned, the five tissues not common are the initials of the root, for which it is currently not possible to obtain specific microarray data. This is another example where our method can truly complement available expression data for the understanding of Arabidopsis development.

We have chosen to validate our model with a series of images taken at arbitrary time points. However, the largest benefit of using GFP reporters and automated image processing for expression analysis is that expression can be monitored in a living organism. In the long term, we plan to further develop our system to be part of an anticipated large scale effort to study transcription factor expression in root development under a variety of environmental conditions. As such, a system to quantify GFP expression values will provide the basis for the computational biology of spatiotemporal gene expression (Bar-Joseph, 2004) and the high resolution elucidation of transcriptional regulatory networks.

ACKNOWLEDGEMENTS

DLM was supported by a training grant from the National Institute of Health. UO is an Alfred P Sloan fellow. This work was supported by National Science Foundation (0209754) to PNB.

REFERENCES

Bar-Joseph, Z. (2004) Analyzing time series gene expression data. *Bioinformatics*, **20**, 2493–2503.

Benfey, P. N. and Scheres, B. (2000) Root development. *Current Biology*, **10**, R813–R815.

Birnbaum, K., Shasha, D. E., Wang, J. Y., Jung, J. W., Lambert, G. M., Galbraith, D. W. and Benfey, P. N. (2003) A gene expression map of the Arabidopsis root. *Science*, **302**, 1956–1960.

Bookstein, F. L. (1989) Principal warps: thin-plate splines and the decomposition of deformations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **11**, 567–585.

Chenyang, X. and Prince, J. L. (1998) Snakes, shapes, and gradient vector flow. *Image Processing, IEEE Transactions on*, **7**, 359–369.

Chudakov, D. M., Lukyanov, S. and Lukyanov, K. A. (2005) Fluorescent proteins as a toolkit for in vivo imaging. *Trends in Biotechnology*, **23**, 605–613.

Dmochowski, I. J., Dmochowski, J. E., Oliveri, P., Davidson, E. H. and Fraser, S. E. (2002) Quantitative imaging of cis-regulatory reporters in living embryos. *PNAS*, **99**, 12895–12900.

Dollar, P. (2006) Matlab TPS (Thin Plate Spline) implementation. *http://vision.ucsd.edu/ pdol-lar/toolbox/doc/index.html*.

Huttenlocher, D. P., Klanderman, G. A. and Rucklidge, W. J. (1993) Comparing images using the Hausdorff distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **15**, 850–863.

Jaeger, J., Surkova, S., Blagov, M., Janssens, H., Kosman, D., Kozlov, K. N., Manu, Myasnikova, E., Vanario-Alonso, C. E., Samsonova, M., Sharp, D. H. and Reinitz, J. (2004) Dynamic control of positional information in the early Drosophila embryo. *Nature*, **430**, 368–371.

Kass, M., Witkin, A. and Terzopoulos, D. (1988) Snakes: Active contour models. *International Journal of Computer Vision*, **1**, 321–331.

Kennedy, J. and Eberhart, R. (1995) Particle swarm optimization. *Neural Networks, 1995. Proceedings, IEEE International Conference on*, **4**, 1942–1948.

Kim, H.-L. (2003) Comparison of oligonucleotide-microarray and serial analysisof gene expression (SAGE) in transcript profiling analysis of megakaryocytes derived from CD34+ cells. *Experimental and Molecular Medicine*, **35**, 460–466.

Kumar, S., Jayaraman, K., Panchanathan, S., Gurunathan, R., Marti-Subirana, A. and Newfeld, S. J. (2002) BEST: A novel computational approach for comparing gene expression patterns from early stages of Drosophila melanogaster development. *Genetics*, **162**, 2037–2047.

Lee, J.-Y., Colinas, J., Wang, J., Mace, D., Ohler, U. and Benfey, P. (2006) Transcriptional and post- transcriptional regulation of transcription factor expression in Arabidopsis roots. *PNAS*, **103**, 6055–6060.

Luc, V. and Pierre, S. (1991) Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, **13**, 583–598.

Maintz, J. B. A. and Viergever, M. A. (1998) A survey of medical image registration. *Medical Image Analysis*, **2**, 1–36.

Ogniewicz, R. and Ilg, M. (1992) Voronoi skeletons: Theory and applications. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR*, pp. 63–69.

Park, P. J., Cao, Y. A., Lee, S. Y., Kim, J.-W., Chang, M. S., Hart, R. and Choi, S. (2004) Current issues for DNA microarrays: platform comparison, double linear amplification, and universal RNA reference. *Journal of Biotechnology*, **112**, 225–245.

Peng, H. and Myers, E. W. (2004) Comparing in situ mRNA expression patterns of Drosophila embryos. In *8th Conf. on Computational Molecular Biology (San Diego, CA, 2004)*, pp. 157–166.

R Development Core Team (2005) *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org. ISBN 3-900051-07-0.

Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S., Lewis, S., Richards, S., Ashburner, M., Hartenstein, V., Celniker, S. and Rubin, G. (2002) Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biology*, **3**, research:0088.1–0088.14.

van Ruissen, F., Ruijter, J., Schaaf, G., Asgharnegad, L., Zwijnenburg, D., Kool, M. and Baas, F. (2005) Evaluation of the similarity of gene expression data estimated with SAGE and Affymetrix genechips. *BMC Genomics*, **6**, 91.

Woods, R. P., Grafton, S. T., Holmes, C. J., Cherry, S. R. and Mazziotta, J. C. (1998) Automated Image Registration: I. general methods and intrasubject, intramodality validation. *Journal of Computer Assisted Tomography*, **22**, 139–152.

Wu, J.-Q. and Pollard, T. D. (2005) Counting cytokinesis proteins globally and locally in fission Yeast. *Science*, **310**, 310–314.

Zitova, B. and Flusser, J. (2003) Image registration methods: a survey. *Image and Vision Computing*, **21**, 977–1000.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e332–e340
doi:10.1093/bioinformatics/btl239

Mutation parameters from DNA sequence data using graph theoretic measures on lineage trees

Reuma Magori-Cohen[1], Yoram Louzoun[1] and Steven H. Kleinstein[2,*]

[1]Math Department, Bar Ilan University, Ramat Gan, Israel, 52900 and [2]Department of Computer Science, Princeton University, Princeton, New Jersey, 08544, USA

ABSTRACT

Motivation: B cells responding to antigenic stimulation can fine-tune their binding properties through a process of affinity maturation composed of somatic hypermutation, affinity-selection and clonal expansion. The mutation rate of the B cell receptor DNA sequence, and the effect of these mutations on affinity and specificity, are of critical importance for understanding immune and autoimmune processes. Unbiased estimates of these properties are currently lacking due to the short time-scales involved and the small numbers of sequences available.

Results: We have developed a bioinformatic method based on a maximum likelihood analysis of phylogenetic lineage trees to estimate the parameters of a B cell clonal expansion model, which includes somatic hypermutation with the possibility of lethal mutations. Lineage trees are created from clonally related B cell receptor DNA sequences. Important links between tree shapes and underlying model parameters are identified using mutual information. Parameters are estimated using a likelihood function based on the joint distribution of several tree shapes, without requiring *a priori* knowledge of the number of generations in the clone (which is not available for rapidly dividing populations *in vivo*). A systematic validation on synthetic trees produced by a mutating birth-death process simulation shows that our estimates are precise and robust to several underlying assumptions. These methods are applied to experimental data from autoimmune mice to demonstrate the existence of hypermutating B cells in an unexpected location in the spleen.

Contact: stevenk@cs.princeton.edu

1 INTRODUCTION

Mutating birth-death processes (MBDPs) are a fundamental component of biology at many different time scales, ranging from evolution of species, through epidemiological evolution of bacteria and other pathogens, to within-host mutation of viruses such as HIV. A special case is the affinity maturation of B cells during an immune response, which is the main focus of this paper (although our methods could be applied to other MBDPs). Affinity maturation normally occurs following the migration of naïve B cells into germinal centers, and the binding of B cell antibody receptors directly to antigens (e.g., molecular determinants on the surface of pathogens) accompanied by a secondary signal resulting from binding helper T cells. Over a three-week period, activated B cells proliferate rapidly and undergo a process of somatic hypermutation whereby point mutations are introduced into the DNA coding for their antibody receptor. According to the theory of clonal selection, B cells with mutations that increase their affinity for antigen gain a proliferative advantage. In this way the average affinity of the population increases over time. For a detailed review of the biology underlying affinity maturation, please see (Wagner and Neuberger 1996).

Despite the significance of MBDPs, methods to estimate underlying parameters from available data are lacking in many cases, particularly when the number of samples is small and the time-scale is short (as is the case for B cell affinity maturation). Population genetic methods have been developed to estimate various evolutionary parameters based on constant rate birth-death models (Nee *et al*., 1994) or coalescent processes (Rosenberg and Nordborg 2002). These approaches assume a large population size (often fixed in the case of coalescent processes) and superimpose a mutation history on a genealogy as a separate step. However, when the number of generations is small the mutation rate impacts the tree topology along with the branch lengths. In addition, population genetic models do not include processes, such as mutation-dependent cell death, that play an important role in B cell affinity maturation.

We previously reported preliminary results on maximum likelihood (ML) methods to estimate the B cell receptor mutation rate based on a small number of B cell lineage tree shapes (Kleinstein *et al*., 2003). Each tree is obtained from a microdissection experiment, which provides a number of clonally related B cell receptor DNA sequences that can be genealogically related to each other using a maximum parsimony algorithm (Clement *et al*., 2000). Many processes, including the hypermutation rate, influence the 'shape' of these clonal trees. Our approach defines a simple MBDP comprising the main biological mechanisms underlying affinity maturation (including clonal expansion with somatic hypermutation and the possibility of lethal mutations), and estimates the parameters of this process. Applying this to biological data from at the Shlomchik lab (Kleinstein *et al*., 2003) suggested, for example, that specific B cells in an autoimmune mouse were undergoing somatic hypermutation in an unexpected area of the spleen. Here we extend and confirm these findings, which have important implications for understanding the etiology of autoimmune diseases such as Lupus.

*To whom correspondence should be addressed.

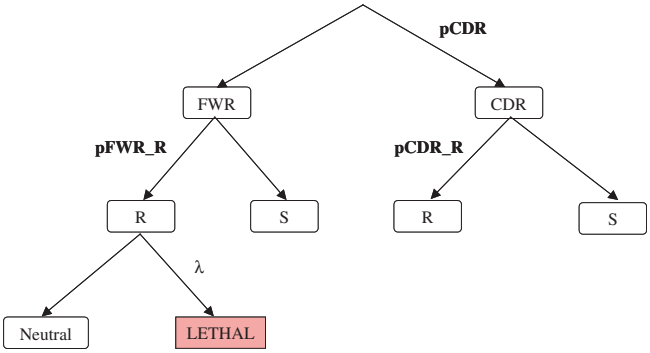


Fig. 1. Mutation decision tree used in MBDP simulations. Mutations occur with a Poisson rate of μ per division. The effect of each mutation depends on whether it falls in the Framework Region (FWR) or the Complementarity Determining Region (CDR) of the receptor gene. The relatively invariant FWRs provide the overall structure of the receptor, and serve to support the more variable CDRs where antigen binding commonly occurs. Following the work of (Shlomchik, Watts *et al*. 1998), each mutation was given a pCDR = 25% chance of being a CDR mutation and a (1-pCDR) = 75% chance of being a FWR mutation. Each mutation also has a pCDR_R = pFWR_R = 75% chance of being a replacement and (1-pCDR_R) = (1-pFWR_R) = 25% chance of being silent. FWR replacement mutations have a probability λ of being lethal. Lethal mutations kill the cell.

The parameters of the MBDP would ideally be estimated by computing the likelihood of producing the set of observed lineage trees (or isomorphic ones) over all realistic parameter values, and finding the combination producing the highest overall probability. However, such full ML methods are expensive to compute. Our previous work sought to develop a more computationally efficient method by summarizing each tree as a set of graph theoretic measures (referred to as tree shapes). In this study we use mutual information to focus the analysis on the most informative shapes. Previous attempts to study tree shapes in the context of evolutionary processes have usually focused on single properties such as the distribution of the number of lineages over time (Nee *et al*., 1994), or branch lengths (Takezaki *et al*., 1995). Analysis of B cell lineage trees up to this point has been limited to statistical approaches based on the average values of individual tree shapes, and qualitative comparison of possible underlying processes (Dunn-Walters *et al*., 2002; Dunn-Walters *et al*., 2004; Mehr *et al*., 2004). The current work differs from these previous attempts in the development of an underlying stochastic model appropriate for B cell clonal expansion, and the quantitative correlation of several tree shapes to allow estimation of multiple parameters of the MBDP model.

2 MODEL AND METHODS

B cell clonal expansion and somatic hypermutation are modeled as a MBDP with multi-type cells and the following three reactions:

- Cell division with average rate of β per generation.
- Stochastic mutation with Poisson rate μ per division.
- Cell death with probability of λ^* per mutation, where $\lambda^* = (1.0 - \text{pCDR}) * \text{pFWR\_R} * \lambda$. pCDR, pFWR and λ are parameters of the mutation decision tree (Figure 1).
- Mutation-independent cell death with rate δ per division.

Table 1. Tree shapes considered in the mutual information analysis. Shapes used in the simulation-based estimate (S) and the analytical estimate (A) are indicated. 'Full' vertices contain observed sequences (seq), while 'empty' ones do not

	Tree Shape Description	S	A
0	Number of full internal vertices	X	
1	Number of empty internal vertices	X	
2	Sequence in full internal vertices		
3	Number of parent-child couples		
4	# of seq. in parent-child couples		X
5	# of seq. in vertices with empty parent		
6	Number of repeated sequences		X
7	Number of internal vertices	X	
8	Number of leaves		
9	Number of seq. in leaves		
10	Number of seq. in internal vertices		
11	Number of vertices	X	
12	Number of sequences at root*	X	X
13	Number of edges		
14	Number of independent mutations		
15	Average number of mutations*	X	X
16	Least common ancestor distance		
17	Replacement-to-Silent ratio, R/(R+S)	X	X

*Note that the simulation estimate defines these measures on unique sequences only.

The MBDP is initiated with a single cell. After d cell generation times, q cells are randomly sampled from the total population of N cells. The set of accumulated mutations in the q sampled cells are used to create a genetic lineage tree as previously described (Kleinstein *et al*., 2003). By construction this tree is correct (i.e., it is a sub-tree of the actual lineage tree). We assume that the experimentally observed trees (created using maximum parsimony on sets of B cell receptor DNA sequences) are correct in the same sense.

The parameters of the MBDP are $\theta = (\beta,\mu,\lambda,\delta,\text{d},q)$. The division rate ($\beta$), assumed to be equal for all cells, defines the time scale of the MBDP and can be set to one through appropriate rescaling. Throughout this study we set $\beta = 1$ and simulate cell division as a deterministic process occurring once during each discrete time step of the simulation. As shown later, relaxing this deterministic assumption has only a minor impact on our estimates. Our estimation methods also assume $\delta = 0$, although we include this parameter in the analytical formula derivations and mutual information analysis. We don't expect this to affect the other parameter estimates (see Summary). The number of sampled cells (q) is included as a parameter to account for the possibility that some observed sequences are repetitions from a single sampled cell due to the particular experimental protocols used. While the simulation-based method we present is insensitive to these potential repetitions, the analytical method assumes a one-to-one correspondence between sampled cells and observed sequences.

Each lineage tree t is summarized by a set of shapes: $S_t = \{s_{t,1}, s_{t,2}, \ldots s_{t,S}\}$, where $s_{t,i}$ is shape i of tree t. The shapes used in this study (defined in Table 1) partially overlap those defined in (Kleinstein *et al*., 2003). We estimate θ by maximizing the likelihood of observing the set of tree shapes S_t over all trees t given the MBDP described above with parameters θ. Expected tree shapes are based on analytical formulae or numerical simulations. To maximize the available information, our approach uses the collective properties of a set of trees assuming the same generative process and equivalent μ and λ, but different d and q for each tree.

We use synthetic data sets to measure the precision of these estimates under different experimental conditions and show that our methods work

with realistic (i.e., small) amounts of experimental data. We first present and validate the methods on synthetic data and then apply them to a set of B cell receptor sequence data derived from microdissection experiments in a mouse model of autoimmune disease. While results from our previous analysis were limited to the mutation rate, here we extend and validate the methods to estimate additional parameters, specifically the lethal mutation frequency and the number of divisions in each clone.

2.1 Simulation of B-cell clonal expansion

The MBDP simulation has been previously described (Kleinstein *et al*., 2003; Kleinstein and Singh 2003). Briefly, it is initiated with a single seeding cell. At each discrete time step (corresponding to one cell generation time), all cells are allowed to divide and die. During each division a Poisson distributed number of mutations occurs with average μ. The impact of a mutation is determined by the mutation decision tree in Figure 1. Cells with lethal mutations are removed after every generation. This process continues until D_{max} generations have passed.

2.2 Mutual information analysis

Mutual information based on Shannon's entropy, one of the central concepts of Information theory, is used to identify tree shapes (both individually and in groups) that provide the maximal information about the underlying MBDP parameters. Shannon's entropy measures the information content of a source X, and is defined as:

$$H(X) = -\sum_{x} \Pr(x) \cdot \log \Pr(x)$$

where Pr(x) is the probability function of the random variable X. Similarly, joint Shannon's entropy is defined as:

$$H(X,Y) = -\sum_{x,y} \Pr(x,y) \log \Pr(x,y)$$

where Pr(x,y) is the joint probability function of the random variables X and Y. These formulas are used to calculate the information content of each combination of tree shapes, denoted as H(Y), as well as combinations of MBDP parameters, denoted as H(X). By definition, the mutual information between the parameters and shapes is:

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

The mutual information measures the information about *X* that is shared by *Y*. In other words, how much of the information about the model parameters X is expressed by the tree shapes Y.

2.3 Analytical estimation method

In the analytical method, optimal model parameters are estimated by minimizing the weighted least squares difference between the observed and expected tree shapes:

$$X(\theta) = \sum_{t=1}^{T} \underset{d}{Min} \left(\sum_{i=1}^{S} \frac{(s^{o}_{t,i} - s^{e}_{t,i}(\theta))^2}{VAR(\{s^{o}_{r,i}\}_{r=1}^{T})} \right)$$

where S is the number of tree shapes considered, $s^{o}_{t,i}$ is the observed value of shape i in tree t, and $s^{e}_{t,i}(\theta)$ is the expected value given the parameters θ. $VAR(\{s^{o}_{r,i}\}_{r=1}^{T})$ is the variance of tree shape *i* calculated over all the observed lineage trees. The minimization of the error $X(\theta)$ takes place in two stages. For each observed tree t, we first minimize the error over all possible numbers of divisions (d), producing an estimate for the number of divisions in the clone that gave rise to the tree (denoted d_t). The overall error is then computed as the sum of the errors for each tree, and this value is minimized to find the optimal values for μ and λ simultaneously. Recall that in this approach we assume the number of sampled cells (q) equals the observed number of sequences as discussed above.

2.4 Simulation-based estimation method

In the simulation-based approach, we begin by estimating λ to be the value where the expected fraction of mutations that are replacements, R/(R+S), is equal to its observed value (I_t) computed over all independent mutations in all trees:

$$\lambda = (I_t - (FWR\_R + CDR\_R))/(FWR\_R*(I_t - 1))$$

where,

$$FWR\_R = (1.0 - pCDR)*pFWR\_R$$
$$CDR\_R = pCDR*pCDR\_R$$

pCDR, pCDR_R and pFWR_R are parameters of the mutation decision tree shown in Figure 1, which describes the distribution of random mutations. These parameters are set to typical values (Shlomchik, Watts *et al*., 1998) although, as we have previously shown, it is easy to estimate them directly for any particular germline sequence of interest (Kleinstein and Singh 2003).

The overall likelihood for producing an experimental data set is the product of the likelihood for each observed tree:

$$L(S_1, S_2 \ldots S_T \mid \theta) = \prod_{t=1}^{T} L(S_t \mid u_t, \theta)$$

where $L(S_t \mid u_t,\theta)$ is the likelihood of observing a tree with shapes S_t given that the microdissection and sequencing produces u_t unique sequences, and assuming an underlying model with parameters θ. This likelihood is also dependent on the number of divisions in the clone, and the number of cells sampled to create the tree. Since neither of these quantities are known with certainty, we sum over all possible values (assuming they are equally likely) to get:

$$L(S_t \mid u_t, \theta) = \sum_{d=1}^{\infty} \sum_{q=u_t}^{q_t} \Pr(S_t, d, q \mid u_t, \theta)$$

where we know that the number of sampled cells included in tree t lies somewhere between the observed number of unique sequences (u_t) and the total number of cells in the microdissection pick (q_t). Of course we cannot simulate an infinite number of divisions in practice, and it is necessary to limit the number of divisions to a computationally reasonable range. This can lead to errors in the estimate due to truncation of the distribution so instead of summing over the entire range we choose the value of d for each tree that maximizes the probability (referred to as d_t):

$$L(S_t \mid u_t, \theta) = \underset{d \le D_{max}}{Max} \sum_{q=u_t}^{q_t} \Pr(S_t, d, q \mid u_t, \theta)$$

The maximum (d_t) is our estimate for the number of generations in the clone that gave rise to tree t. The upper bound on the number of divisions in the simulation (D_{max}) is set to a value that is thought to upper bound the clonal expansion size (and is computationally feasible). $\Pr(S_t,d,q \mid u_t,\theta)$ is estimated using Monte Carlo simulations with parameters θ by:

$$\Pr(S_t, d, q \mid u_t, \theta) = \frac{E(S_t, d, q, u_t)}{\sum_{d,q} U(d, q, u_t)}$$

Here, $U(d,q,u_t)$ is the number of simulated trees with u_t unique sequences after *d* divisions and randomly sampling *q* cells. Among these trees, $E(S_t,d,q,u_t)$ is the number that are also equivalent to the observed tree in all shapes specified in Table 1 (i.e., the simulated tree can be summarized by S_t).

To calculate $U(d,q,u_t)$ and $E(S_t,d,q,u_t)$, an expanding B cell clone is simulated beginning from a single cell as described in Section 2.1. After each division, *q* cells are randomly sampled from the population and a lineage tree is created as previously described (Kleinstein *et al*., 2003). If the number of unique sequences in this tree is u_t, then $U(d,q,u_t)$ is incremented by one, where d is the number of divisions so far. If the simulated tree also has shape S_t, then $E(S_t,d,q,u_t)$ is incremented by one.

After running the simulation *i* times, the likelihood of producing each of the observed trees is determined. We used $i = 128{,}000$ independent simulation runs to calculate the likelihood at each value of the mutation rate since more runs did not provide additional accuracy (data not shown). The overall mutation rate μ is estimated by maximizing the likelihood using golden section search.

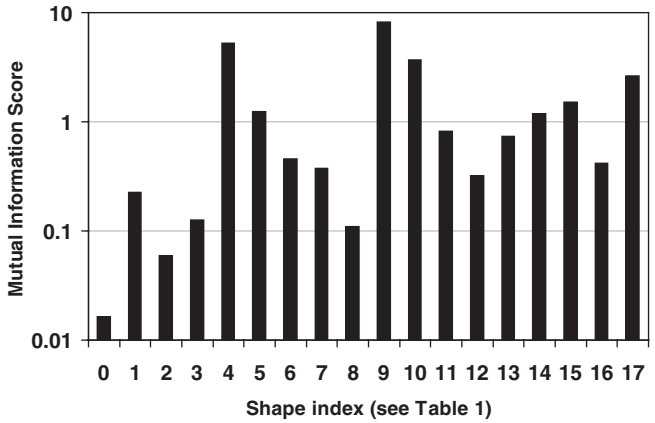


Fig. 2. Contribution of individual tree shapes to estimation of simulation parameters θ. Shape triplets were ordered by their mutual information with the MBDP parameters, and weighted by an exponentially decreasing function. Bar heights are the sums of the weighted frequency of each shape. Synthetic data included 1000 trees for every combination of: $\mu = \{0.1, 0.3, 0.5, 0.7, 0.9\}, \lambda = \{0.0, 0.25, 0.50\}, \delta = \{0.0, 0.2, 0.4\}, \mathrm{d} = \{0,1,\ldots,15\}$, with $q = \min(\mathrm{N},10)$.

3 RESULTS

3.1 Optimal tree shapes

Individual tree shapes can reflect different aspects of the underlying biological process, but some contain similar information. The relationship between model parameters (θ) and resulting tree shapes can be highly non-linear so that classical linear regression measures do not properly represent the contribution of particular shapes to the estimate of θ. We use mutual information based on Shannon's entropy to determine which shapes contain the most information about θ. By varying the set of shapes included in S_t and measuring the mutual information $I(S_t,\theta)$, we can estimate how much information is conveyed by S_t about θ. We define the most informative set of shapes as the one with the highest mutual information.

To determine the most informative set of tree shapes, we computed the mutual information between S_t and $\theta = (\mu,\lambda,\delta,\mathrm{d},q)$ for a set of synthetic trees produced by simulating the MBDP for a range of realistic parameter values (see Figure 2 caption). An equivalent analysis was done under the assumption that q is known, using an extended shape vector $S_t^* = \{S_t,q_t\}$ and removing q from θ, and similar results were obtained (data not shown). We considered all possible combinations of three tree shapes from the set listed in Table 1. The optimal shape triplet for the simultaneous estimate of θ is composed of: (1) the ratio of Replacement (R) to Silent (S) mutations, (2) the number of sequences in parent-child couples, and (3) the number of sequences in the leaves of the tree. However, several other triplets have similar mutual information. We developed a scoring system to identify individual tree shapes that repeatedly appear in high mutual information triplets. First, all shape triplets are sorted by their mutual information content and given an exponentially decreasing weight. The score for individual shapes (i.e., components of S_t) is calculated by summing the weights of all triplets that contain it. As shown in Figure 2, the highest scoring individual shapes include those in the optimal triplet. Note that some shapes score highly, but are equivalent to other shapes so that there is no benefit to using them simultaneously, while other measures may have a relatively low score but are required to complete a good triplet. Quadruplets can be done in a similar way.

As described in the following section, we have derived analytical equations to approximate the expected values of several high scoring tree shapes in Figure 2 (indicated in Table 1). These shapes are used to summarize each lineage tree in the analytical estimation approach. The simulation-based estimate does not use many of these shapes since they would require assuming a one-to-one correspondence between sampled cells and observed sequences. We can avoid this assumption by restricting the set of valid shapes to those that do not depend on the number of repeated sequences. The limited number of such topological tree shapes allows us to use them all (up to closely related ones).

3.2 Analytical method results

As indicated in Table 1, five tree shapes were used in the error function to estimate θ. The following sections outline formulas for the expected values for each of these tree shapes. The full development cannot be included here due to space limitations. Note that these derivations include mutation-independent cell death with rate δ per division.

Average mutations per sequence. Consider a cell that has undergone d divisions and accumulated m mutations. The number of such cells surviving (e.g., accumulating no lethal mutations) is: $\alpha^{\mathrm{d}}(1\text{-}\lambda_1)^{\mathrm{m}}$ where λ_1 = (1-pCDR) x pFWR_R x λ is the overall probability that a mutation will be lethal (see Figure 1), and $\alpha = 2e^{-\delta}$. The probability of having m mutations after d generations is a Poisson process with an average of μd. Thus, after d divisions the expected number of cells with m mutations that are still alive is:

$$\alpha^d(1-\lambda_1)^{\mathrm{m}}e^{-\mu d}\frac{(\mu d)^m}{m!}$$

The number of live cells (N) after d divisions is calculated by summing over all possible numbers of mutations:

$$N = \sum_m \alpha^d(1-\lambda_1)^{\mathrm{m}}e^{-\mu d}\frac{(\mu d)^m}{m!}$$

Thus, the expected number of mutations per sequence is:

$$\begin{aligned}\mathrm{M} &= \frac{1}{N}\sum_m \alpha^d(1-\lambda_1)^{\mathrm{m}}e^{-\mu d}\frac{(\mu d)^m}{m!}m \\ &= \mathrm{E}[\text{Poisson process with mean}(1-\lambda_1)\mu d] \\ &= (1-\lambda_1)\mu d\end{aligned}$$

After simplifying, we find that M is simply the expected branch length in the absence of lethal mutations (μd) multiplied by the probability of cell survival ($1-\lambda_1$).

Number of unique sequences. The expected number of unique sequences (u) in a random sample of q cells can be computed by the probability that a given sequence is different from all others. The number of unique sequences is:

$$\sum_{i=1}^{q}\Pr(\textit{sequence i} \neq \textit{sequence } j; j < i)$$

This can be approximated by:

$$\sum_{i=1}^{q}\Pr(i \neq 1)\Pr(i \neq 2 \mid i \neq 1)\ldots\Pr(i \neq i-1 \mid i \neq 1..i-2)$$

The probability that two random sequences are different is:

$$1 - X, \text{ where } X = \frac{(e^{-2\mu}((2\cdot\alpha\cdot e^{-2\mu})^d - 1))/(2\cdot\alpha\cdot e^{-2\mu} - 1)}{(2\cdot\alpha^d - 1)/(2\cdot\alpha - 1)}$$

The contribution of the conditional probability varies between 1 and $2^{1\text{-}i}$, so that the number of unique sequences is bounded between:

$$\mathrm{u} = \sum_{i=1}^{q}\prod_{j=1}^{i-1}(1-X)^i \text{ and } \mathrm{u} = \sum_{i=1}^{q}\prod_{j=1}^{i-1}\left(1-\frac{X}{2^j}\right)^i.$$

Averaging these values gives an excellent fit to the number of unique sequences in simulated trees (data not shown).

Average sequences at the root. Each sequence appearing at the root of a lineage tree represents a cell that has undergone d divisions without accumulating any mutations. The probability of this occurring for a single cell is $e^{-\mu d}$. Such cells will be enriched in the population due to the death of cells that accumulate lethal mutations. The fraction of cells in the root is thus the expected number of unmutated cells divided by the total number of surviving cells:

$$\frac{\alpha^d e^{-\mu d}}{\alpha^d(1-\lambda_1\mu)^d} = \left(\frac{e^{-\mu}}{(1-\lambda_1\mu)}\right)^d$$

Multiplying this fraction by the number of sampled cells (q_t) in any particular clonal tree (t) gives the number of sequences expected to be present at the root (R_t):

$$R_t = q_t \times \left(\frac{e^{-\mu}}{(1-\lambda_1\mu)}\right)^d$$

Sequences in parent-child nodes. The probability for a pair of sequences to appear as a parent-child couple in a tree can be computed as the probability that in two nearby branches of the actual lineage tree, one branch is mutated while the other is not. When the tree is collapsed to create the equivalent of the maximum parsimony tree, the sequence in the unmutated branch becomes the parent of the sequence in the mutated branch. The probability to find two such sequences is:

$$\begin{aligned}
&\sum_{i=1}^{d} e^{-\mu i}(1-e^{-\mu i})\cdot(2\cdot e^{-\lambda\mu-\delta})^{2i-2}(2\cdot e^{-\lambda\mu-\delta})^{d-i}\cdot\frac{q(q-1)}{N(N-1)}\\
&\quad\times 2^{(1-\mu-5\delta)} = \frac{q(q-1)\cdot(2\cdot e^{-\lambda\mu-\delta})^{d-2}}{N(N-1)}\cdot 2^{(1-\mu-5\delta)}\cdot\\
&\quad\times\left[\sum_{i=1}^{d}(2e^{-\mu-\lambda\mu-\delta})^i - \sum_{i=1}^{d}(2e^{-2\mu-\lambda\mu-\delta})^i\right]\\
&\quad= \frac{q(q-1)\cdot(2\cdot e^{-\lambda\mu-\delta})^{-2}}{(N-1)}\cdot 2^{(1-\mu-5\delta)}\cdot\\
&\quad\times\left[\frac{2e^{-\mu-\lambda\mu-\delta}(2^d e^{-\mu d-\lambda\mu d-\delta d}-1)}{2e^{-\mu-\lambda\mu-\delta}-1}\right.\\
&\quad\left.-\frac{2e^{-2\mu-\lambda\mu-\delta}(2^d e^{-2\mu d-\lambda\mu d-\delta d}-1)}{2e^{-2\mu-\lambda\mu-\delta}-1}\right]
\end{aligned}$$

Estimates using the analytical method

The direct application of the analytical error minimization method, using the tree shapes in Table 1 with expected value computations described above, provides unbiased estimates of μ and λ as tested on synthetic data sets (Figure 3). Looking at the error surface, we initially found that the minimum was indeed at the correct location, but the error function was practically flat in the direction of λ, resulting in a large variance in the estimate of the frequency of lethal mutations. This was improved by increasing the weight of R/(R+S) in the error function (Figures 3 and 4). This tree shapes is directly affected by λ, but not by μ.

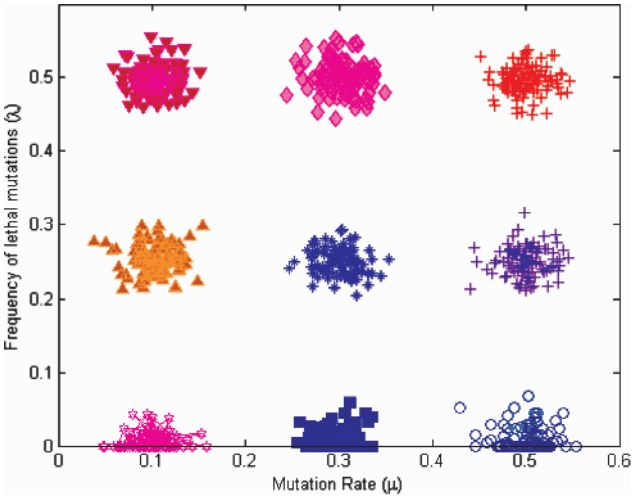


Fig. 3. Estimate of the mutation rate (μ) and lethal frequency (λ) using the analytical method. Each synthetic data set contains T = 100 trees with $q = 10$ cells. Data was created for all combinations of $\mu = (0.1,0.3,0.5)$ and $\lambda = (0.00,0.25,0.50)$. Each point is the estimated value on one synthetic data set. Clusters are centered on the actual values. Note the Replacement-to-Silent ratio was given a 10-fold weight increase in the error function.

3.3 Simulation-based method results

As discussed previously (and shown in Figure 4 for the analytical method), the likelihood is relatively flat as a function of λ suggesting that individual trees contain little information to estimate this parameter. Although we found it was possible to simultaneously estimate μ and λ using the analytical approach, the inherent noise in the simulation estimate, resulting from the limited number of simulations used to estimate each likelihood, makes that strategy infeasible here. Thus, we employ a two-step approach. First the lethal mutation frequency (λ) is estimated by considering the fraction all independent mutations that are replacements, R/(R+S). Since silent mutations, which do not change the amino acid coded for, cannot be lethal to the cell, this fraction provides a direct, albeit noisy, signal to estimate the lethal mutation frequency. The mutation rate μ is then estimated by comparing the expected and observed tree shapes (not including R/(R+S) which was used to estimate λ), assuming a particular value of λ (either known or estimated). Previous estimates of μ were based on educated guesses about the number of generations. Our method simultaneously estimates μ along with the number of divisions giving rise to each tree (further denoted d_t).

In contrast to the analytical method, the simulation-based method is robust to the assumption that repeated sequences represent unique cells. First, this method uses only shapes that do not depend on the precise number of repeated sequences. Second, all possible values for the number of sequences (q) are considered in the likelihood

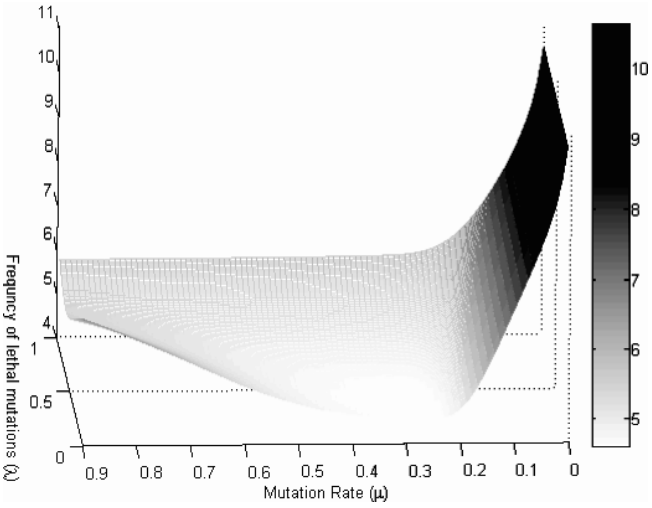


Fig. 4. Representative error landscape for analytical estimates based on a synthetic data set with 100 trees for μ and λ where the actual values are $\mu = 0.3$ and $\lambda = 0.25$. The minimum is at the correct position, but the error landscape is very flat in the λ direction. Note the Replacement-to-Silent ratio was given a 10-fold weight increase in the error function.

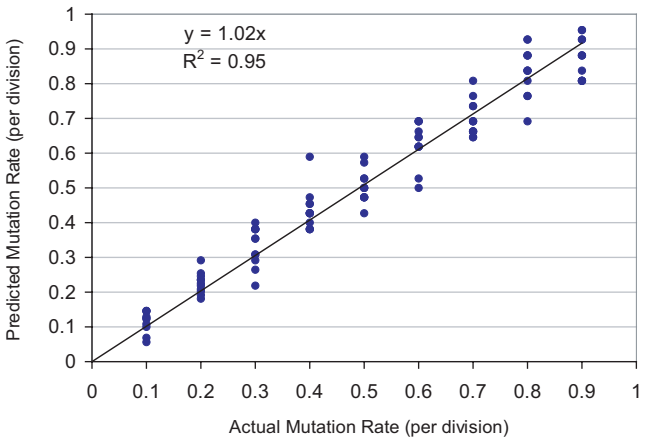


Fig. 5. Estimating the mutation rate with the simulation method when the frequency of lethal mutations (λ) is known. Individual points show the results on each synthetic data set, consisting of $T = 50$ clonal trees each with $q = 5$ cells. For each tree the sampling time was randomly distributed between 5 and 10 generations (as is the case with all our synthetic data sets except where indicated). At least 7 synthetic data sets were produced at each mutation rate investigated. $\lambda = 0.5$ and $D_{max} = 10$ for all likelihood evaluations.

computation. Another advantage of this approach is that, while the analytical formulas estimate the expected value of each tree shape independently, the simulation-based method numerically estimates the joint distribution of all shapes.

The mutation rate and lethal frequency

In previous work we proposed a simulation-based method to estimate the mutation rate (μ) from a set of lineage trees (Kleinstein *et al*., 2003). However, this method produced a biased estimate. The method presented here differs from that approach by explicitly summing over all possible numbers of sequences in the observed tree (q_t). Furthermore, instead of directly sampling the observed number of unique sequences from the simulated tree, we condition our likelihood on this value. This ensures that singletons (i.e., trees containing only a single unique sequence), whose true frequency is impossible to estimate experimentally, do not unduly influence the results.

We validate the improved method by estimating μ from synthetic data sets where the actual mutation rate is known. The frequency of lethal mutations (λ) has been previously estimated for some well-studied responses (Shlomchik 1990), and as a first step we consider the case where this parameter is known. From Figure 5, it is clear that this new method is unbiased and converges to the actual value of μ. In addition, the variance in the estimate of the likelihood decrease as the sample size grows (data not shown). Even when the number of trees and sequences is small (as is the case for the actual experimental data), our method provides a reasonable estimate of the mutation rate (Figure 6).

Like the simulation used to estimate the likelihood function, the synthetic data sets assume that cell division is synchronous so that all cells in a clone (giving rise to a single tree) have undergone the same number of divisions. To test whether our method is sensitive to this assumption, we generated synthetic data sets with asynchronous division using the discrete time-step approach developed in

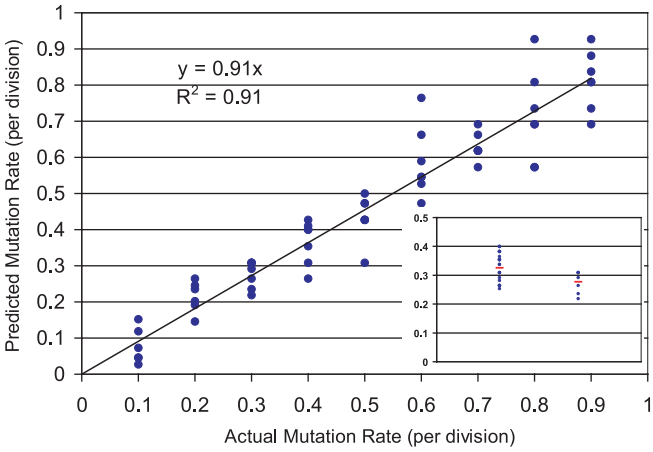


Fig. 6. Estimating the mutation rate for limited data with the simulation method when the frequency of lethal mutations (λ) is known. Synthetic data sets were created that had the same number of trees and sequences as the experimentally derived autoimmune response data. The inset shows that our method is not overly sensitive to the assumption of synchronous division. Synthetic data sets were created using an asynchronous division model (left) and a synchronous model (right). The simulation method (which uses a synchronous model) was applied to estimate the mutation rate whose actual value is $\mu = 0.3$ per division. $D_{max} = 12$ for all likelihood evaluations.

(Kleinstein and Singh 2001). In this model the time between divisions is Poisson distributed with average value $\beta = \ln(2)$ leading to a doubling time of one, which is equivalent to the synchronous model with discrete time steps. We then applied our estimation method (which still used synchronous division) to these data. The estimated mutation rates were close whether or not the synthetic data used synchronous or asynchronous division (Figure 6 inset).

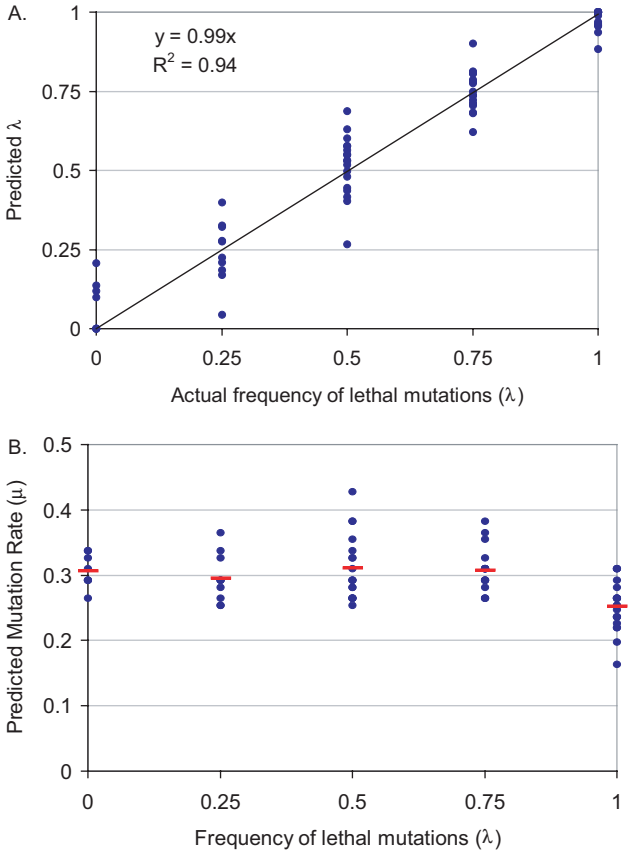


Fig. 7. Predicting the mutation rate with the simulation method when the frequency of lethal mutations (λ) is unknown. Synthetic data sets contain $T = 50$ trees with $q = 5$ cells each and $\mu = 0.3$. (A) The predicted value of lambda for individual synthetic data sets. (B) Predictions of the mutation rate for synthetic data sets created with the indicated value of λ. The underestimation at $\lambda = 1$ (which is biologically unrealistic in any case) is due to this being a hard upper bound. $D_{max} = 12$ for all likelihood evaluations.

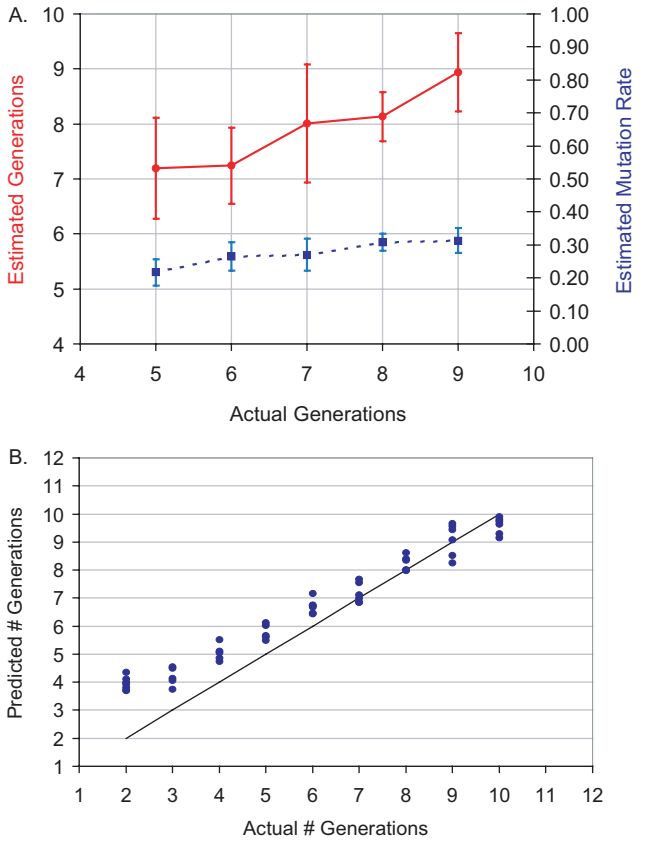


Fig. 8. Predicting the average number of generations using the simulation-based method. (A) Individual points are average results on at least 7 synthetic data sets consisting of $T = 50$ clonal trees each with $q = 5$ cells randomly sampled from a clone with the indicated number of generations. Error bars indicate one standard deviation. These results used the actual value for λ, and estimated the mutation rate μ (dotted line) and the average number of divisions for all clones in the data set (solid line). (B) The predicted number of generations when the mutation rate is known. Each point is the result from one synthetic data set. When the actual number of generations approaches the number of simulated generations used to calculate the likelihood (D_{max}), we can also underestimate the number of divisions. This is easily corrected by raising this bound at the expense of computation time. $D_{max} = 12$ for all likelihood evaluations.

We next consider how well we can estimate θ when the frequency of lethal mutations (λ) is unknown. Although we find that the estimate of λ itself is quite noisy (Figure 7a), this does not greatly impact the estimate of μ (Figure 7b). In fact, when $\lambda = 0.5$ (a value estimated in previous work), the standard deviation in the prediction of μ increases only from 0.06 to 0.07 per division for the estimates with known and unknown λ respectively. This is true even though the estimate of λ itself has a standard deviation of 0.1 for the reasons previously discussed for the analytical estimate.

The number of generations

To check if the number of divisions maximizing the likelihood for each tree (d_t) is a reasonable estimate of the actual number of divisions in the clone giving rise to tree t, we generated synthetic data sets where all the clones had the same fixed number of generations. Multiple data sets were created spanning the range from 5 through 9 generations. For each of these data sets, the mutation rate was estimated (assuming the actual value for λ is known), and the average number of divisions among all the clones in the data set was predicted. This prediction was positively correlated with the actual number of generations (Figure 8a). It was accurate when the number of divisions was high. When the number of divisions was below ~7 this method overestimates the average number of divisions, which leads to an underestimate in the predicted mutation rate for these clones. This linkage makes intuitive sense since a clone that has longer to mutate can achieve the same frequency of mutations with a lower mutation rate. The direction of causality is suggested by the observation that the method overestimates the number of divisions at low generation numbers even when the mutation rate μ is known (Figure 8b). In these cases our method provides a lower bound on the mutation rate.

3.4 Analysis of autoimmune data

We applied our methods to estimate mutation parameters from a set of experimentally derived lineage trees collected from autoimmune

mice (William *et al*., 2002). Details of the tree shapes are described in (Kleinstein *et al*., 2003) and (Magori-Cohen *et al*., 2006). From these data we estimate using the simulation-based method that ~55% of FWR replacement mutations are lethal (i.e., $\lambda = 0.55$), that the clones have undergone ~5 divisions on average, and that the mutation rate is ≥ 0.26 generation^{-1}, corresponding to approximately $0.26/340 = 7.6\text{x}10^{-4}$ base-pair^{-1} generation^{-1}. The analytical method produces a similar estimate of the mutation rate. We can conclude that these clones are undergoing hypermutation at a rate consistent with a 'classic' immune response (McKean *et al*., 1984; Wabl *et al*., 1985; Wabl *et al*., 1987). This is an important (and surprising) result since these cells were microdissected from the T zone-red pulp border rather than germinal centers, where hypermutation is thought to be restricted. This has significant implications for understanding the etiology of autoimmune diseases such as Lupus.

4 SUMMARY

Estimating mutation properties is a key element in much of the theory dealing with evolution of cells or species. Currently existing methods assume a very high number of generations, and often a large population size. These assumptions do not apply in many systems, such as the short-term evolution of viruses in a human host or B cell affinity maturation during an immune response (the main focus of this paper). We have developed a MBDP simulation to model the B cell affinity maturation process, along with two ML methods for estimating the mutation parameters (including somatic hypermutation rate, lethal mutation frequency and the number of generations). The input to our methods consists of a set of maximum parsimony lineage trees generated from experimentally observed groups of clonally related B cell receptor DNA sequences. The correctness of the maximum parsimony reconstructions was tested on synthetic data sets and found to be precise for over 98% of trees. Our methods are based on an initial selection of the most informative tree shapes (based on mutual information). In the first method, we derive analytical estimates for the expected value of each tree shape given a set of parameters, and compare these with the observed shapes using weighted least squares. The second method, based on numerical simulations of the underlying MBDP, was developed to cover cases where repeated sequences could be artifacts of the specific experimental protocols employed. Although limited by its high computational requirements, it has the additional advantage of estimating the full joint distribution of tree shapes instead of estimating the expected value of each shape individually. The analytical method can be viewed as a first rapid approximation to this full distribution estimate. The validity of these methods is verified using synthetic data sets. Our methods provide unbiased estimates of the mutation rate, the lethal mutation frequency and the age (in cell generations) of each tree when the number of generation is higher than seven, and a lower bound for younger trees, even for cases where the amount of data is limited.

Preliminary results suggest that our current approach fails to estimate the rate of mutation-independent cell death. We have generated data sets similar to the one used in the analysis above and included mutation-independent cell death with a rate of δ per division, and attempted to estimate δ. We found that the ML curves were too flat in the direction of δ to provide any insight (data not shown). This result is not surprising since cell death is equivalent to missing a full branch in the cell-sampling step, which is a frequent occurrence in any case due to the small number of cells sampled to create each tree. Consequently, we expect the MBDP parameter estimates will be unaffected by assuming $\delta = 0$.

It is possible to extend our MBDP model to include other biological processes, such as selection. Negative selection is currently included in the analysis in the form of lethal mutations, but there is currently no positive selection (for a discussion of why this is not critical for the particular experimental data analyzed, see (Kleinstein *et al*., 2003)). Positive selection in its simplest form could be described using a model with two populations with equal mutation rates, but one dividing (or dying) faster than the other. In such a simple model the ratio between the variance in the number of mutations per sequence in the trees and their average would be greater than one. On the other hand, this ratio is not sensitive to the occurrence of lethal mutations and we find in our synthetic data that, as expected, the average is equal to the mean (Magori-Cohen *et al*., 2006). Significant deviations from one would suggest the presence of positive selection, and require that an appropriate model be built for the relevant system. Note that evolutionary relationships (Goldman 1994), or non-homogeneous sampling might also result in ratios higher than one.

We expect that the low frequency of tree reconstruction errors (less than 1.5%) will have a limited effect on the final parameter estimates since these combine multiple structural elements with elements taken purely from the sequences (e.g., R/(R+S)). Another assumption that could impact our methods validity is that of synchronous division. While we expect this assumption will be approximately true of the *in vivo* data as a result of the small microdissections and short time-scales being considered, we have used synthetic data to show that relaxing this assumption does not significantly affect our results (Figure 6 inset).

In summary, we have developed a rapid, systematic measure of mutation parameters from small sets of DNA sequence data, based on a limited set of lineage tree shapes deemed most relevant to the underlying process. We have further provided a methodology based on comparison with synthetic data to test the limits of its applicability.

ACKNOWLEDGEMENTS

The work of Y.L. and R.M.C. was covered by BSF grant 2003328 and the EU 6th framework co3 pathfinder. S.H.K. was supported in part by NSF IGERT grant DGE-9972930.

REFERENCES

Clement,M. *et al*. (2000) ''TCS: a computer program to estimate gene genealogies.'' *Mol. Ecol*., **9**(10), 1657–9.

Dunn-Walters,D.K. *et al*. (2002) ''The dynamics of germinal centre selection as measured by graph-theoretical analysis of mutational lineage trees.'' *Dev. Immunol*., **9**(4), 233–43.

Dunn-Walters,D.K. *et al*. (2004) ''Immune system learning and memory quantified by graphical analysis of B-lymphocyte phylogenetic trees.'' *Biosystems*., **76**(13), 141–55.

Goldman,N. (1994) ''Variance to mean ratio, R(t), for poisson processes on phylogenetic trees.'' *Mol. Phylogenet. Evol*., **3**(3), 230–9.

Kleinstein,S.H. *et al*. (2003) ''Estimating hypermutation rates from clonal tree data.'' *J. Immunol*., **171**(9), 4639–49.

Kleinstein,S.H. and Singh,J.P. (2001) ''Toward quantitative simulation of germinal center dynamics: biological and modeling insights from experimental validation.'' *J. Theor. Biol.*, **211**(3), 253–75.

Kleinstein,S.H. and Singh,J.P. (2003) ''Why are there so few key mutant clones? The influence of stochastic selection and blocking on affinity maturation in the germinal center.'' *Int Immunol.*, **15**(7), 871–84.

Magori-Cohen,R. *et al.* (2006) , http://www.cs.princeton.edu/~stevenk/trees.

McKean,D. *et al.* (1984) ''Generation of antibody diversity in the immune response of BALB/c mice to influenza virus hemagglutinin.'' *Proc. Natl Acad Sci. USA*, **81**(10), 3180–4.

Mehr,R. *et al.* (2004) ''Analysis of mutational lineage trees from sites of primary and secondary Ig gene diversification in rabbits and chickens.'' *J. Immunol.*, **172**(8), 4790–6.

Nee,S. *et al.* (1994) ''Extinction rates can be estimated from molecular phylogenies.'' *Philos. Trans. R Soc. Lond. B Biol. Sci.*, **344**(1307), 77–82.

Rosenberg,N.A. and Nordborg,M. (2002) ''Genealogical trees, coalescent theory and the analysis of genetic polymorphisms.'' *Nat. Rev. Genet.*, **3**(5), 380–90.

Shlomchik,M.J., Litwin,S. and Weigert,M.G. (1990) The influence of somatic hypermutation on clonal expansion. *rogress in Immunology.* In *Proceedings of the Seventh International Congress of Immunology,* **7**, 415.

Shlomchik,M.J. *et al.* (1998) ''Clone: a Monte-Carlo computer simulation of B cell clonal expansion, somatic mutation, and antigen-driven selection.'' *Curr. Top Microbiol. Immunol.*, **229**, 173–97.

Takezaki,N. *et al.* (1995) ''Phylogenetic test of the molecular clock and linearized trees.'' *Mol. Biol. Evol.*, **12**(5), 823–33.

Wabl,M. *et al.* (1985) ''Hypermutation at the immunoglobulin heavy chain locus in a pre-B-cell line.'' *Proc. Natl. Acad Sci. USA*, **82**(2), 479–82.

Wabl,M. *et al.* (1987) ''Measurements of mutation rates in B lymphocytes.'' *Immunol Rev.*, **96**, 91–107.

Wagner,S.D. and Neuberger,M.S. (1996) ''Somatic hypermutation of immunoglobulin genes.'' *Annu. Rev. Immunol.*, **14**, 441–57.

William,J. *et al.* (2002) ''Evolution of autoantibody responses via somatic hypermutation outside of germinal centers.'' *Science*, **297**(5589), 2066–70.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e341–e349
doi:10.1093/bioinformatics/btl263

Indel seeds for homology search

Denise Mak[1,*], Yevgeniy Gelfand[2] and Gary Benson[3,†]

[1]Graduate Program in Bioinformatics, Boston University, Boston, MA 02215, [2]Lab for Biocomputing and Informatics, Boston University, Boston, MA 02215 and [3]Department of Computer Science, Department of Biology, Boston University, Boston, MA 02215

ABSTRACT

We are interested in detecting homologous genomic DNA sequences with the goal of locating approximate inverted, interspersed, and tandem repeats. Standard search techniques start by detecting small matching parts, called *seeds*, between a query sequence and database sequences. Contiguous seed models have existed for many years. Recently, spaced seeds were shown to be more sensitive than contiguous seeds without increasing the random hit rate. To determine the superiority of one seed model over another, a model of homologous sequence alignment must be chosen. Previous studies evaluating spaced and contiguous seeds have assumed that matches and mismatches occur within these alignments, but not insertions and deletions (indels). This is perhaps appropriate when searching for protein coding sequences (<5% of the human genome), but is inappropriate when looking for repeats in the majority of genomic sequence where indels are common. In this paper, we assume a model of homologous sequence alignment which includes indels and we describe a new seed model, called *indel seeds*, which explicitly allows indels. We present a *waiting time* formula for computing the sensitivity of an indel seed and show that indel seeds significantly outperform contiguous and spaced seeds when homologies include indels. We discuss the practical aspect of using indel seeds and finally we present results from a search for inverted repeats in the dog genome using both indel and spaced seeds.

Contact: dyfmak@bu.edu

1 INTRODUCTION

Standard heuristic algorithms for homology search in biological sequences (Pearson and Lipman, 1988; Altschul *et al*., 1990, 1997; Kent, 2002) utilize a two step approach. In the search step, short words from the query sequence, called *seeds* are paired with all matching words in sequences from a target database. The pairing can be done efficiently by first indexing the database words. In the confirmation step, each database match, called a *hit*, is tested to see if it is part of an extended region with homology to the query sequence. Hits are tested using alignment or some approximation to alignment.

An important element in successful homology detection programs is the choice of a good seed. A short seed increases the probability of finding a hit within a homologous region, but also increases the number of random hits in non-homologous regions and thereby increases the running time. A long seed reduces the number of random hits, but also reduces the probability of hitting a homologous region. In practice, a trade-off is made between sensitivity (the probability of hitting a homologous region) and excessive running time caused by too many random hits. BLAT (Kent, 2002) for example, is well designed for DNA regions sharing very high identity ($\geq$95%), allowing the use of long seeds or multiple seeds which provide both excellent sensitivity and very low probability of random hits.

In the last few years, much interest and research has focused on what are called *spaced seeds* (Ma *et al*., 2002; Buhler and Sun, 2005; Keich *et al*., 2004; Brejova *et al*., 2004; Choi and Zhang, 2004; Choi *et al*., 2004; Sun and Buhler, 2004; Xu *et al*., 2004; Brejova *et al*., 2005; Noe and Kucherov, 2005) because they increase sensitivity *without* simultaneously increasing the number of random hits. Patternhunter (Ma *et al*., 2002) was the first general purpose program to utilize spaced seeds. While a standard contiguous seed is a short word or substring drawn from the query sequence, a spaced seed is a non-contiguous *subsequence*, that is, it consists of a number of explicit positions which must match separated by ''don't care'' positions where the query and the homologous region may or may not match. A recent extension of this concept, implemented in YASS (Noe and Kucherov, 2005), is called a ''transition constrained seed'' and requires that mismatches in the ''don't care'' positions must be of the transition type (A to G, or C to T) rather than transversions.

All the recent work on spaced seeds assumes that mutational differences in homologous regions consist solely of substitutions (mismatches) or that insertions and deletions (indels), if present, are widely spaced. While it can be argued that this is a valid assumption for designing seeds to find homologous protein coding regions, it is *not* valid for homology search in sequence where indels are tolerated, such as promoters and non-coding repeats. One of us (Benson) has been involved for several years in the development of software (Tandem Repeats Finder-TRF Benson, 1999 and Inverted Repeats Finder-IRF Warburton *et al*., 2004) for the detection of approximate DNA repeats. These repeats usually occur within or contain non-coding sequence and along with substitution mutations, they exhibit numerous indel mutations as well. For example, in roughly 10% of the IRs found in the human genome using IRF (Warburton *et al*., 2004), with arm lengths between 50 bp and 100 bp and arm separation below 500 000 bp, the frequency of indels between the left and right arms exceeds 8%. In a 50 bp repeat, this means at least 4 indel positions, with a limited possibility of wide spacing. This

*To whom correspondence should be addressed.

†This research was supported in part by NSF grant DBI-0413462.

estimate of indel frequency is low since the detection method uses *k*-tuple matching (an alternate name for contiguous seeds). Switching to spaced seeds would improve the sensitivity of IRF, but developing a seed model that is sensitive to the existence of indels would improve the sensitivity even more.

In this paper, we introduce the idea of *indel seeds*. As with spaced seeds, an indel seed is a subsequence of the query which must match exactly to produce a hit and the separating ''don't care'' positions may or may not match. The difference is that some separating positions can be of variable size to allow for insertions or deletions between the matching parts. In order to accommodate indels, we model alignments between homologous regions with a four character alphabet: {match, mismatch, insertion in query, insertion in target} and use a Markov chain to specify transition probabilities between these characters. We show how to compute the sensitivity of indel seeds under this model using a ''waiting time'' (Aki *et al*., 1984) calculation. We determine optimal seeds under several match/mismatch/indel configurations and show that indel seeds are more sensitive than contiguous or spaced seeds (with equivalent random hit rates) even when the indel frequency is one third the mismatch frequency. We discuss how to use indel seeds in practice, and finally, we present the results of searches for inverted repeat homologies in the dog genome using both indel and spaced seeds.

This paper is organized as follows. In section 2 we define our model of alignments for homologous regions that include insertions and deletions. In section 3, we define indel seeds. In section 4 we explain how indel seeds are used in practice and we discuss the random hit rate of indel seeds. In section 5 we show how to compute the sensitivity of indel seeds. In section 6 we compare the sensitivity of indel, contiguous, and spaced seeds. Finally, in section 7 we present the results of our search for inverted repeats in the dog genome.

2 HOMOLOGY MODEL

Two sequences are homologous if they have a common evolutionary ancestor. After the sequences diverge due to duplication or speciation, they typically undergo a variety of mutational events which, over time, transform them into different sequences. Depending on the evolutionary pressures, the sequences may change rapidly or remain similar over many millions of years. Homology detection tools exploit the remaining similarities to identify sequences that are homologous. Optimal seed selection is based on the homology model, which includes 1) the set of mutations that are presumed to transform the sequences, 2) the frequency of those mutations, and 3) the length of alignments of homologous regions.

A simple homology model and the one most frequently studied (Ma *et al*., 2002; Keich *et al*., 2004; Buhler and Sun, 2005; Xu *et al*., 2004; Choi and Zhang, 2004; Sun and Buhler, 2004; Brejova *et al*., 2005) allows only substitution mutations. In this model, alignments consist solely of matches and mismatches, and are represented by bit strings where a 1 indicates a match and a 0 indicates a mismatch or substitution. For example:

```
A C G T G C G T A A T T T C G
A C C A G C T T T A T T C C G
1 1 0 0 1 1 0 1 0 1 1 1 0 1 1
```

Two common variations assume 1) that the match and mismatch frequencies are independent and identically distributed (iid) across the alignment or 2) that the frequencies are the result of a Markov chain or hidden Markov model where every third position has a higher substitution frequency in order to reflect the higher variability in third codon position in protein coding gene sequences (Buhler and Sun, 2005; Brejova *et al*., 2004). Another model introduced by Noé and Kucherov (Noe and Kucherov, 2004) represents alignments with a ternary alphabet rather than bit strings in order to accommodate the higher frequency of transition substitutions in DNA evolution. In this case, one character represents a match, one a transition mismatch (A to G or T to C) and one a transversion mismatch (everything else).

Indel Model. Our new homology model includes insertion and deletion events as well as substitution. An alignment is represented by a string (hereafter called the *representative string*) over the following four character alphabet:

0 – mismatch;
1 – match;
2 – insertion into database sequence (deletion from query sequence);
3 – insertion into query sequence (deletion from database sequence).

For example:

```
                Query  A C - G T G C G T A A T T T C G
             Database  A C C G A G C - T - - T T T T G
Representative String  1 1 2 1 0 1 1 3 1 3 3 1 1 1 0 1
```

Normalized alignment length. In the match-mismatch homology model, the position *x* in a representative string (alignment) is the same as the position in the query. In the indel homology model, the alignment and the query will differ in length due to insertions into the database sequence. Since homology search proceeds from the query, we show (in Section 5) how to compute the probability of finding a seed hit *relative to the length of the query*. We therefore use the following notion:

DEFINITION. *The normalized length of a representative string of length n is the number of '0', '1', and '3' characters contained in positions 1 to n. Normalized position k in a representative string is the position of the kth character from the set {0, 1, 3}, counting from the left. For example, the following representative strings all have normalized length 5 and in the middle string, pattern 11 occurs at normalized position 5:*

11101 10122211 1331221

From this definition, occurrence of the pattern 111 at normalized position 3 corresponds to an *infinite* number of representative strings because any number of 2's can precede the pattern. This set of strings can be specified as 2* 111, where we use the regular expression 2* to denote zero or more 2's.

Representative string probabilities. Mutation frequencies are described by a first-order Markov chain in order to avoid the occurrence of the character pairs ''2 3'' or ''3 2'' in a representative string. We exclude these pairs because consecutive indels, one in each sequence, are typically excluded by the choice of alignment parameters: the penalty for a single mismatch is usually less than the combined penalty for two individual indels. For the remainder of

this paper, we use the first-order Markov chain specified in the 4 x 4 transition matrix below:

From: \ To:	0	1	2	3
0	p_0	p_1	p_g	p_g
1	p_0	p_1	p_g	p_g
2	p_0^*	p_1^*	p_g	0
3	p_0^*	p_1^*	0	p_g

where p_g stands for the probability of a gap symbol ('2' or '3') which we assume to be the same in the query and the database, and $p_i^* = p_i + p_g\,(p_i/(p_0 + p_1))$ for $i = 0, 1$ represents the proportional distribution of p_g to the characters 0 and 1. Other probability distributions are possible. We assume that the Markov process is stationary so that the probability of a given digit at any position reflects the equilibrium distribution $\pi = (\pi_0, \pi_1, \pi_2, \pi_3)$. However, the same is **not** true of any *normalized position*. In particular, this implies that the probability of a '1' at a given normalized position x depends on x. In actuality, the situation is simpler than that as the following theorem states. This fact is important for the sensitivity formula derived in Section 5.

THEOREM 1. *In a representative string, the probability of a '1' at the first normalized position is*

$$\pi_1 + \pi_2\left(\frac{p_1}{p_0 + p_1}\right)$$

and at all other normalized positions is

$$p_1^* = p_1 + p_1\left(\frac{p_g}{p_0 + p_1}\right)$$

(Proof omitted.)

3 SEED MODEL

The seed model defines characteristics of homologous alignments that will be recognized by the homology detection program. The model is specified in terms of the alphabet of representative strings and wildcard symbols. A *contiguous seed* is a string of 1's denoting successive matches in the alignment, for example 11111 or 1^k where k in this case is 5. Ma *et al*. (Ma *et al*., 2002) described a *spaced seed* which is a string, beginning and ending with a 1 and containing 1's and $*$'s where $*$ is a wildcard that denotes either 1 (match) or 0 (mismatch) at that position. For example, 11$**$11 indicates that a match or mismatch can occur at each of positions 3 and 4. Thus, 11$**$11 actually specifies four distinct patterns in the representative string alphabet: 111111, 111011, 110111, 110011.

DEFINITION. *An indel seed is a string beginning and ending with a 1 and containing 1's, $*$'s, and X's where 1 represents a match, $*$ is a wildcard that denotes either 1 (match) or 0 (mismatch), and X is a wildcard that indicates zero or one characters from the set {0, 1, 2, 3}, i.e., either a match, mismatch, insertion into the database or insertion into the query. Consecutive X's can represent any pair of numeric digits except ''2 3'' and ''3 2''.*

For example, 11XX1 with two wild-cards, permits indels of size zero, one, or two. This seed specifies the following 19 pattern strings:

111, 1101, 1111, 1121, 1131, 11001, 11111, 11221, 11331, 11011,

11021, 11031, 11101, 11121, 11131, 11201, 11211, 11301, 11311

As another example, the indel seed 1X1$*$1 permits indels of size zero or one and specifies the following 10 pattern strings:

1101, 1111, 10101, 10111, 11101,
1111, 12101, 12111, 13101, 13111

4 USING INDEL SEEDS

In the following we outline how the indel seed 11XX11 is used in practice. The method generalizes to seeds with different numbers of indel positions. To process the query, Q, of length n, at a typical index $i \in [6, n]$, we extract *three* patterns because the size of the insertion in the query can be zero, one, or two:

$$Q[i-5, i-4]Q[i-1, i], \quad Q[i-4, i-3]Q[i-1, i],$$

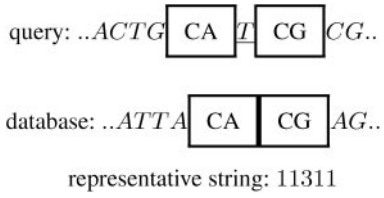
and

$$Q[i-3, i-2]Q[i-1, i].$$

For example, if Q = ACTGCATCGCG..., then the patterns extracted at position 9 (underlined) are:

Note that the rightmost pair of characters is the same in all three patterns. The differences are in the position of the leftmost pair, separated by either two, one or zero characters from the rightmost pair. Because of edge effects, at query index $i = 4$ there is one pattern to extract and at query index $i = 5$ there are two. Each pattern is treated as a single four character string, in other words, *the spacing between the pairs is ignored*. We process the database sequences in the same way, again selecting three patterns for a typical index because the insertion in the database sequence can also be zero, one, or two.

Determining what constitutes a match. In the simplest approach, a match occurs whenever *any* pattern from a group of three in the query matches *any* pattern in a group of three from the database. The difference in spacing is not considered relevant for a match and is interpreted as an indel. For example:



Random Hit Rate. Because there are multiple chances to match for each query pattern, the random hit rate is based on the number of 1's in the seed and the number of comparisons. With the four-letter DNA alphabet, when we assume that each letter occurs with equal frequency, the probability of a random match is approximately

$$9 \cdot (1/4)^4$$

Table 1. Four possible comparisons between the query and database sequences in a restricted comparison strategy and the representative strings they detect

		Query patterns:		
		11	1_1	1_ _1
Database patterns:	11	-	131	-
	1_1	121	-	1301, 1311, 1031, 1131
	1_ _1	-	1201, 1211, 1021, 1121	-

The exponent 4 comes from the four characters in the query and database strings (the four 1's in the seed). The factor 9 comes from the fact that each query position has 3 patterns and each pattern has 3 chances of matching to a database position. The probability is actually somewhat less because the patterns at any one position are not independent (as in the example, the last two characters are the same).

Restricted comparison strategy. Note that there is flexibility in deciding which patterns from the query and database should be compared to detect matches and this flexibility can be used to reduce the random hit rate if the sensitivity of the seed model is not seriously impaired. For example, a more restrictive comparison approach would allow matching only between two patterns whose insertion spacing differs by one. This might be desireable when the seed is long and there is a high probability of finding an indel within the seed length. For this approach, the probability of a random match is

$$4 \cdot (1/4)^4$$

where the factor of 4 comes from the four possible comparisons between the three query patterns and three database patterns as in Table 1.

5 SEED SENSITIVITY

We calculate seed sensitivity using a *waiting time* formula (Aki *et al*., 1984), which is a general technique for calculating the probability of observing a given event, in a randomly generated string, by the occurrence of the *k*th character. Waiting time formulas are equivalent to other methods for calculating seed sensitivity (Keich *et al*., 2004; Buhler and Sun, 2005; Brejova *et al*., 2004; Choi and Zhang, 2004).

Waiting Time Formula. For clarity of presentation, the full version of this manuscript begins by deriving formulas for spaced seeds. Due to space restrictions, those formulas have been omitted. We assume that we are given an indel seed which specifies a set of patterns in the representative string alphabet. From this set, we eliminate all patterns that contain, as a substring, another pattern from the set. This can happen because an 'X' in the seed allows some of the patterns to be shorter than others. We work with this reduced set of patterns. The following terms are used:

- T: The set of patterns specified by the seed, after eliminating those containing other patterns as substrings.
- L_i: The *normalized* length of $pattern_i$.
- w_i^*: The probability of $pattern_i$ which is dependent on its normalized position. (By Theorem 1, that probability is the same everywhere except when the initial '1' occurs at the first normalized position.)
- $w_i[s]$: The probability of the suffix of $pattern_i$ *following* normalized position s, $s \in [1..L_i - 1]$ which is independent of its position.
- $V_i[s]$: The set of patterns from T that can overlap the prefix of $pattern_i$ when they occur at normalized position s, $s \in [1..L_i - 1]$.
- P ($pattern_i : x$): The probability, across all representative strings, of $pattern_i$ being the first occurrence of *any* pattern at *normalized* position x.
- $S(seed : x)$: The *sensitivity* of the seed, *i.e*., the *cumulative* probability, across all representative strings, of a first occurrence of any of the set of patterns in T at all *normalized* positions between 1 and x. ($S(seed : x) = \sum_{k=1}^{x} (\sum_{a \in T} P(pattern_a : k))$).

We are interested in calculating the probability of first occurrences of every $pattern_i \in T$ at every normalized string position x, where first occurrence means a string ends with $pattern_i$ at position x and the string prefix up to normalized position $x - 1$ contains no other pattern in T. The seed sensitivity is the sum of these probabilities. The general scheme is as follows. For each $pattern_i$, and each normalized position x, calculate the probability of all representative strings with normalized length x that end with $pattern_i$ (which is just w_i^*). Then, subtract the probability of all strings in this set which contain an earlier first occurrence of any pattern in T.

To determine earlier first occurrences, $pattern_i$ is compared with every other pattern (including itself) for overlapping and non-overlapping positions. All patterns are non-overlapping when they occur at $k \leq x - L_i$. The probability of strings which both end with $pattern_i$ and contain one of these earlier occurrences is:

$$w_i^* \cdot \sum_{k=1}^{x-L_i} \left(\sum_{a \in T} P(pattern_a : k) \right) = w_i^* \cdot S(seed : x - L_i)$$

The following determines the probability of the intial 1 in $pattern_i$, and thus the value of w_i^*, when $pattern_i$ follows an earlier occurrence of some pattern (rather than being the first occurrence of any pattern in the string).

COROLLARY 2. *The probability of a '1' at any normalized position* following *an earlier occurrence of any pattern is* p_1^*.

(Proof omitted.)

At each position $j > x - L_i$ some subset of the patterns, $V_i[L_i - (x - j)] \subseteq T$, occurring at j, is consistent with an overlap of $pattern_i$ at position x. The probability of these patterns is multiplied by the probability of the suffix of $pattern_i$ that extends past the overlap:

$$\sum_{j=x-L_i+1}^{x-1} \left[\left(\sum_{a \in V_i[s]} P(pattern_a : j) \right) w_i[s] \right]$$

where $s = L_i - (x - j)$
The probability of a first occurrence of $pattern_i$ at position x is then,

$$P(pattern_i : x) = w_i^*(1 - S(seed : x - L_i)) - \sum_{j=x-L_i+1}^{x-1} \left[\left(\sum_{a \in V_i[s]} P(pattern_a : j) \right) w_i[s] \right] \quad (1)$$

where $s = L_i - (x - j)$

THEOREM 3. *The time complexity for computing the cumulative sensitivity of an indel seed,* $S(seed : n)$, *is* $O(nl \mid T \mid)$ *where n is the*

length of the homology region, l is the length of the seed, and $|T|$ is the number of strings in the set of patterns specified by the seed.

PROOF. Consider a single $pattern_i$. From the formula for $P(pattern_i : x)$, for every location between 1 and n in the representative string, $pattern_i$ computes one add and one multiply for the term $w_i^*(1 - S(seed : x - L_i))$ and $L_i - 1$ multiplications in the term $\sum_{j=x-L_i+1}^{x-1} [(\sum_{a \in V_i[s]} P(pattern_a : j))w_i[s]]$, where $L_i \leq l$ and $s = L_i - (x - j)$. It suffices to show that the summation terms, $\sum_{a \in V_i[s]} P(pattern_a : j)$, each of which represents the probability for an overlap set, $V_i[s]$, can all be precomputed in time proportional to the number of patterns $|T|$. We sketch the proof here.

An Aho-Corasick tree (AC tree) of all patterns (constructed in time $O(l|T|)$) can be used to determine the overlap set for every position in any $pattern_i$. The chain of failure links from the **end** of any $pattern_j$ specifies the patterns and positions for which $pattern_j$ belongs to an overlap set. Over the entire AC tree, these chains form another tree rooted at the single child of the AC root (indicating that a '1' has been read, since all patterns begin and end with a '1'). Each bifurcating node in this tree as well as each parent node of a leaf (if not bifurcating) represents a single overlap set. The number of these nodes is $< 2|T|$ and they can be identified in a preprocessing step. For a given representative string position x, once the probabilities of all patterns at x have been computed, the probability of an overlap set at x (specified by a bifurcating or leaf parent node in the failure chain tree) can be computed by adding the probabilities of the node's children. This takes time $O(|T|)$ for all overlap sets if the additions are computed in a post-order traversal. ■

For an indel seed, the number of patterns $|T|$ depends on the number and location of the X's. An indel seed with a single X (like those we test in the next section) produces

$$4 \cdot 2^{|*|} + 2^{|*|} = 5 \cdot 2^{|*|}$$

seeds where $|*|$ is the number of match-mismatch wildcards in the seed. The first term specifies the seeds where 'X' holds a character and the second term where 'X' does not.

6 COMPARING SEED SENSITIVITIES

We compared the sensitivity of the three seed classes: contiguous, spaced, and indel. For the homology model, we chose several match/mismatch/indel probability configurations that reflect increasing ratios of indels to mismatches. These ratios are not uncommon in human IRs detected by the Inverted Repeats Finder in an exploration of that genome (Warburton *et al*., 2004). The majority of those human IRs had lengths between 30 and 200 base pairs (bp). For our comparison, we chose two query lengths from this range: 64 and 100. The lower value was chosen to conform with earlier studies of spaced seeds.

For each model, we compared the sensitivity of seeds with equivalent random hit rates. For contiguous and spaced seeds, the random hit rate is $(1/4)^k$ where the exponent, referred to here as *equivalent weight*, is the number of '1's in the seed, assuming equal probability of the letters in the DNA alphabet. For indel seeds, the random hit rate depends on the number and length of the indel positions in the seed as discussed in Section 4. For this analysis, we used indel seeds with a single X (which requires the selection of two strings at each position in the query and database sequences), so the random hit rate is $\sim 4 \cdot (1/4)^k = (1/4)^{k-1}$, that is, the equivalent weight is *one less* than the number of '1's in the seed. For each homology model, we tested seeds with weights equivalent to 8, 9, 10, 11, and 12.

Table 2. Three possible comparisons under a restricted comparison strategy for indel seeds and the representative strings they detect

		Query patterns: 11	1_1
Database patterns:	11	-	131
	1_1	121	101, 111

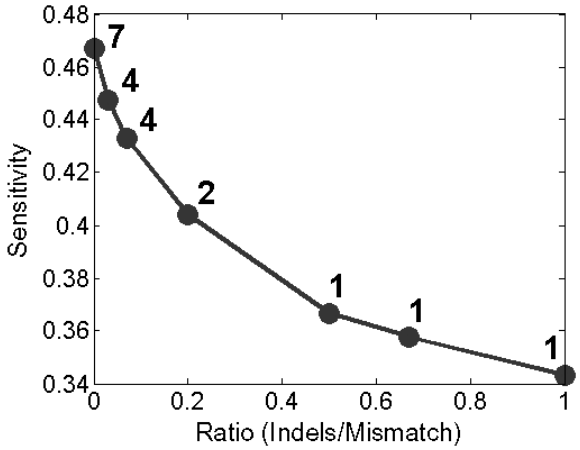


Fig. 1. Increasing the indel to mismatch ratio decreases the number of wildcards in the best spaced seeds (numbers next to data points) as well as the sensitivity. Match probability is 70%.

We also evaluated a restricted comparison strategy for indel seeds. Let the equivalent weight of an indel seed in the unrestricted comparison strategy, be its *nominal* weight. Our restricted comparison strategy allows three comparisons between the query and database at each location, as shown in Table 2. The true random hit rate is $\sim 3 \cdot (1/4)^k$, which is lower than the nominal rate. Out of necessity, we compared restricted comparison indel seeds with spaced seeds having a weight equivalent to the indel seed's nominal weight, *i.e*., with spaced seeds that are expected to have more random hits.

Results. First, we note that contiguous seeds did more poorly than either spaced or indel seeds in almost all comparisons and their sensitivities are not shown.

Second, we observe two trends with spaced seeds: their sensitivity drops significantly as the ratio of indels to mismatches increases and the effectiveness of added match-mismatch wildcards also declines. Figure 1 graphs the sensitivity of the best spaced seed (equivalent weight = 11 and query length = 64) in several configurations, holding the match probability at 70% and raising the indel/mismatch ratio from zero to one. At each datapoint, the integer specifies the number of wildcards in the best seed. Across the range of ratios, sensitivity drops roughly 13% while there is a rapid decline from 7 wildcards in the PatternHunter seed at a ratio of zero, to 2 wildcards at a ratio of 0.2 and one wildcard at a ratio of 0.5. The trend suggests that a contiguous seed (zero wildcards)

Table 3. Sensitivities and optimal seeds at query lengths 64 and 100. Higher sensitivities are shown in bold font. Homology models are specified as (match, mismatch, database insert, query insert). In this table, match probability was held constant at 70%. Last number at top of each column is the ratio of indel probability to mismatch probability. W is equivalent weight, L is query length. Length 64 sensitivities marked with a star (⋆) were obtained with a different seed than the one shown

Model		(70, 25, 2.5, 2.5); 0.2		(70, 20, 5, 5); 0.5		(70, 18, 6, 6); 0.66		(70, 15, 7.5, 7.5); 1	
W	L	Spaced seed	Indel seed	Spaced seed	Indel seed	Spaced seed	Indel seed	Spaced seed	Indel seed
8		**111∗11∗111**	11∗11X1∗1111	**11111∗111**	11111X1111	111∗11111	**11111X1111**	11111∗111	**11111X1111**
	64	**0.799248**	0.775944	**0.770976**	0.766357	0.760703	**0.772317**	0.743978	**0.781979**
	100	**0.927726**	0.915175	**0.909198**	0.906417	0.902463	**0.910265**	0.891094	**0.916372**
9		**1111∗11∗111**	111∗11X11∗111	**11111∗1111**	111111∗1111	11111X1111	**11111X1111**	1111∗11111	**111111X1111**
	64	**0.666672**	0.645569	**0.628948**	0.625259	0.617831	**0.631623**	0.599931	**0.642072**
	100	**0.836377**	0.821216	**0.802802**	0.800117	0.793025	**0.805638**	0.776882	**0.814576**
10		**11111∗11∗111**	111∗11X11∗1∗111	**1111∗111111**	**1111∗111X1111**	111111∗1111	**111111X11111**	111111∗1111	**111111X11111**
	64	**0.526124**	0.511339⋆	**0.488697**	0.487703⋆	0.477954	**0.493522**	0.460946	**0.503144**
	100	**0.710387**	0.698711	0.669123	**0.670198**	0.657549	**0.674724**	0.638893	**0.684828**
11		**1111∗111∗1111**	1111∗11X111∗111	111111∗11111	**11111∗111X1111**	111111∗11111	**1111111X11111**	111111∗11111	**1111111X11111**
	64	**0.404111**	0.394377	0.366688	**0.368235**	0.357618	**0.371836**	0.343322	**0.379957**
	100	**0.578892**	0.570552	0.531516	**0.536510**	0.520300	**0.538502**	0.502391	**0.548347**
12		**11111∗111∗1111**	1111∗1∗11X111∗111	11111∗111∗1111	**111111X111∗1111**	1111111∗11111	**111111X1111111**	1111111∗11111	**1111111X111111**
	64	**0.301648**	0.294727⋆	0.268688	**0.271181**	0.260742	**0.273027**	0.249229	**0.279409**
	100	**0.453702**	0.449161	0.409587	**0.413569**	0.396584	**0.413819**	0.380711	**0.422375**

Table 4. Sensitivities and optimal seeds at query lengths 64 and 100. Higher sensitivities are shown in bold font. Homology models are specified as (match, mismatch, database insert, query insert). In this table, match probabilities are higher than in table 3. Last number at top of each column is the ratio of indel probability to mismatch probability. W is equivalent weight, L is query length

Model		(75, 10, 7.5, 7.5); 1.5		(80, 10, 5, 5); 1		(85, 15, 2.5, 2.5); 0.33	
W	L	Spaced seed	Indel seed	Spaced seed	Indel seed	Spaced seed	Indel seed
8		1111∗1111	**1111X11111**	11111∗111	**11111X1111**	**11111∗111**	1111∗11X111
	64	0.876635	**0.922186**	0.968706	**0.980779**	**0.978886**	0.978114
	100	0.966614	**0.984287**	0.996388	**0.998369**	**0.998096**	0.998057
9		1111∗11111	**111111X1111**	1111∗11111	**1111X111111**	**111∗11∗1111**	**111∗11X11∗111**
	64	0.773066	**0.837862**	0.924164	**0.948027**	**0.943899**	0.943214
	100	0.911442	**0.949127**	0.985147	**0.992035**	0.991157	**0.991239**
10		111111∗1111	**11111X111111**	111111∗1111	**11111X111111**	**11111∗11∗111**	**1111X111∗1111**
	64	0.652459	**0.729514**	0.855497	**0.892439**	**0.886191**	0.890837
	100	0.824181	**0.883914**	0.958263	**0.974373**	0.972386	**0.974289**
11		111111∗11111	**1111111X11111**	111111∗11111	**11111X1111111**	1111∗111∗1111	**111∗111X11∗1111**
	64	0.533202	**0.612365**	0.769472	**0.817654**	0.813606	**0.820006**
	100	0.716664	**0.792340**	0.911418	**0.940129**	0.938616	**0.943307**
12		11111∗1111111	**1111111X111111**	1111111∗11111	**1111111X111111**	11111∗111∗1111	**1111∗11X111∗1111**
	64	0.423409	**0.497833**	0.673449	**0.727906**	0.727865	**0.736757**
	100	0.600568	**0.683390**	0.844450	**0.885424**	0.886638	**0.894515**

would eventually outperform a spaced seed. We observe this for match probability 75% and ratio 1.5 for equivalent weights 8 and 9 (not shown). Our intuition for this trend is the following. Spaced seeds can never match across an indel, so indel characters have the effect of chopping representative strings into smaller pieces. Within these smaller pieces, a longer spaced seed, with more wildcards, has fewer positions in which to match and so has lower sensitivity overall.

Third, with increasing indel to mismatch ratios, indel seeds outperform spaced seeds. Tables 3 and 4 give the sensitivities of the spaced and indel seeds in the homology models tested. In both tables, ''winning seeds'' and their sensitivities are shown in a bold font. In Table 3 the match probability is held constant at 70%. When the indel to mismatch ratio is 0.2, spaced seeds are superior by around 1%. At ratio 0.5, indel seeds for the higher equivalent weights do better and at ratio 0.66 and above, indel seeds are clearly superior. For example, at ratio 1 and equivalent weights 10 and 11, the gain in sensitivity for the indel seed is 4.6%.

Table 4 shows a mix of match probabilities. Except for ratio 0.33 at the lowest equivalent weights, the indel seeds are superior. For example at 80% matching, ratio 1, and equivalent weight 11, the gain in sensitivity for the indel seed is almost 3%, and at 75% matching, ratio 1.5, it is 7.6%.

Table 5. Sensitivity error correction. Sensitivities of optimal seeds from tables 3 and 4 calculated on 100 000 pairs of sequence strings, generated according to the indicated homology model as described in the text, for query length 64. Higher sensitivities are shown in bold font. Sensitivity difference is the gain in sensitivity over the standard calculation. Homology models are specified as (match, mismatch, database insert, query insert). Last number at the top of each column is the ratio of indel probability to mismatch probability. W is equivalent weight, L is query length. The seed marked with a star (⋆) is optimal at query length 64 in table 3 but is not shown in that table

	Model	(70, 20, 5, 5); 0.5		(70, 15, 7.5, 7.5) ;1		(75, 10, 7.5, 7.5) ;1.5	
W	L	Spaced seed	Indel seed	Spaced seed	Indel seed	Spaced seed	Indel seed
10		1111∗111111	**11111X11111⋆**	111111∗1111	**111111X11111**	111111∗1111	**11111X111111**
	64	0.53586	**0.55074**	0.53357	**0.59884**	0.72930	**0.81476**
sensitivity difference		+0.047163	+0.063037	+0.072624	+0.095696	+0.076841	+0.085246
11		111111∗11111	**11111∗111X1111**	111111∗11111	**1111111X11111**	111111∗11111	**1111111X11111**
	64	0.40279	**0.43109**	0.40295	**0.46348**	0.61086	**0.70799**
sensitivity difference		+0.036102	+0.062855	+0.059628	+0.083523	+0.077658	+0.095625
12		11111∗111∗1111	**111111X111∗1111**	1111111∗11111	**1111111X111111**	11111∗1111111	**1111111X111111**
	64	0.30066	**0.32093**	0.29561	**0.34708**	0.48975	**0.58839**
sensitivity difference		+0.031972	+0.049749	+0.046381	+0.067671	+0.066341	+0.090557

Finally, the restricted comparison strategy, while not as sensitive as unrestricted comparison, outperforms spaced seeds with *higher* random hit rates. At match probability 80%, ratio 1, and query length 100, the best restricted comparison indel seeds slightly outperformed the best spaced seeds at nominal weights 11 and 12 (data not shown). We observed the same results at match probability 75%, ratio 1.5, and query length 100 for nominal weights 9 through 12, with a gain of 3% in sensitivity at nominal weight 12.

Seed sensitivity as a lower bound. Seed sensitivity calculations actually underestimate the sensitivity of indel and spaced seeds when using an indel homology model for the following reason. When indels are allowed, sometimes more than one optimal alignment is possible. For example, the following are two optimal alignments for the same pair of sequences.

```
                  Query  A T C T G G A T T G C
               Database  A T A T - G A T T C C
Representative String  1 1 0 1 3 1 1 1 1 0 1

                  Query  A T C T G G A T T G C
               Database  A T A T G - A T T C C
Representative String  1 1 0 1 1 3 1 1 1 0 1
```

Note that in the first alignment, the indel seed, 11X111, does not occur, but in the second alignment it does. Thus we can not always classify the representative string from the first alignment as excluding the seed 11x111. Note also that for some alignments that do not contain a seed, there may be suboptimal alignments that do. For these reasons, our calculated values for indel and spaced seed sensitivities are lower bounds on the true sensitivities. We have estimated the error for several of the optimal seeds from Tables 3 and 4 using the following procedure.

We generated 100 000 random representative strings according to one of our Markov chain homology models, and then for each representative string, generated a pair of sequence strings that match the alignment. For the sequence strings, we assumed equal probabilities for the letters A, C, G, and T. We then exhaustively checked if any alignment, including any suboptimal alignment, of the two sequences contained the seed being tested. We analyzed three different homology models for query length 64. The seed sensitivities for the representative strings were nearly identical to our calculated values (data not shown). The seed sensitivities for the sequence pairs are shown in Table 5. We found a gain in sensitivity of $\sim$ 3.1–9.5% for the seeds over the standard calculation, but in every case the gain was higher for the indel seed than the spaced seed.

7 INVERTED REPEAT SEARCH IN THE DOG GENOME

We tested the ability of indel and spaced seeds to find inverted repeats (IRs) in a modified version of the Inverted Repeats Finder (IRF) (Warburton *et al*., 2004). The best seeds (equivalent weight = 12) were chosen from the homology model for 70% matching and indel to mismatch ratio = 1.0 (Table 3). The program was tested on the first 33 000 000 bases of chromosome I of the dog genome (Lindblad-Toh *et al*., 2005). Typically when using IRF, we first mask known interspersed and tandem repeats so they are not reported as IRs. Interspersed repeats from the same family appear as IRs if they are coincidentally inserted in reverse orientation, tandem repeats from the same family act similarly, In addition, some common tandem repeats such as ATATATAT look like IRs. Sequence masked for interspersed repeats was obtained from the UCSC genome browser website (which uses RepeatMasker) and was additionally masked for tandem repeats using the Tandem Repeats Database (Benson, 2005). Roughly 39% of the sequence was masked and did not participate in IR detection. IRF scans through a sequence, recording seeds at each position by linking locations of identical seeds. The seed at the current forward-most position finds its ''hits'' by scanning backward through the list of its reverse complement. A user specified maximum lookback distance sets the maximum spacer length between IR arms. For each hit, an alignment is computed and the program reports those alignments which score above a user specified minimum. Parameters used were: alignment scoring: match = 2, mismatch = −5, indel = −5; minimum alignment score = 40; lookback = 15,000,000. Testing was performed on a 2.8GHz PC with 2GB RAM. Table 6 shows the results.

Table 6. Results of IR search in dog genome using indel and spaced seeds

Seed	Repeats	Unique Repeats	Avg. Length (uniques)	Middle 50% Range (uniques)		Hits	Time (Minutes)
				Indel to Mismatch Ratio	Match %		
Spaced	21 365	228	56.6	0.2–1.1	82%–89%	38 717 975	23.7
Indel	21 752	614	34.3	1.0–3.1	89%–93%	41 063 612	28.6

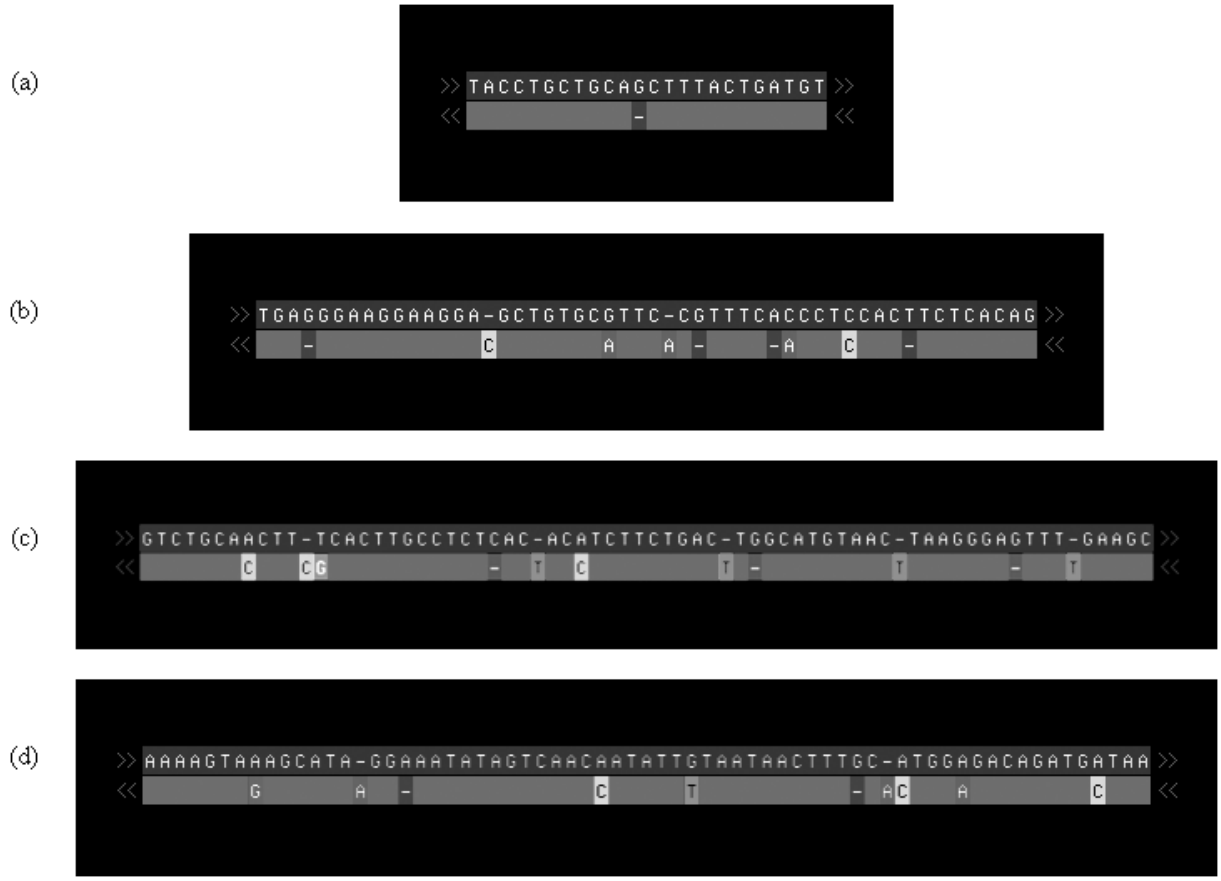


Fig. 2. Examples of IRs found in the dog genome. In each part, upper is the left arm and lower is the right arm. Only mutational differences are shown in the right arm. a) typical of 53 repeats with a single indel found with the **indel** seed but not possible to find with the spaced seed, left arm 24 bp, arms 14.5 Mb apart, % match = 95.8, b) found uniquely by **indel** seed, left arm 50 bp, arms 7.6 Mb apart, % match = 82.7, indel to mismatch ratio = 2.0, c) found uniquely by **indel** seed, left arm 65 bp, arms 12 kb apart, % match = 84.2, ratio = 2.66, d) found uniquely by **spaced** seed, left arm 65 bp, arms 4.9 Mb apart, % match = 85.1, ratio = 0.66.

First notice that the indel and spaced seed are each able to find repeats that the other misses (the unique repeats column). Excluding these, both seeds found 21 138 repeats. Since the predicted sensitivity of these seeds is between 38% and 42% for the homology model used, we expect that many repeats were not found. Next, notice that the indel seed finds roughly 270% more unique repeats than the spaced seed. The unique repeats found by the indel seed are 1) significantly shorter as a group than the unique repeats found by the the spaced seed, and 2) have higher indel to mismatch ratios (averages agree, but the middle 50% range is shown to better illustrate the variation). Surprisingly, the match percentage is generally higher for the indel seed than for the spaced seed. In fact, the spaced seed found no unique repeats with match percentage greater than 93% while the indel seed found 53. Each of the 53 looks like the one shown in figure 2 (part a) with a single indel in the middle. These are impossible to find with the spaced seed. But it should be clear that there exist other repeats with an indel or mismatch offset from the center which were not found by either of the seeds tested here.

Finally, note that the indel seed produces 2 345 637 or $\sim 6\%$ more hits than the spaced seed. And because it does more alignments, the run with the indel seed took 20% longer. The higher number of hits does not appear to be the cause for the higher number of repeats detected by the indel seed (recall the 53 repeats described above). Rather, we believe the higher number of hits is caused by hit *clumping* within repeats which is magnified for the indel seed because multiple hits are examined at each sequence location. Two lines of evidence support this idea. First, runs on randomly generated sequence show less than 1% difference in the number of hits for

the seeds used here. Random sequences rarely contain IRs as long or longer than these seeds (13 and 14 characters) and so hit clumping is not expected to occur. Second, in a trial with the same dog sequence but a lookback of only 1 000 000, a similar increase in the indel hits was observed. Masking the IRs found and reruning the sequence eliminated roughly 1/4 of the excess hits for the indel seed. To test if the remainder were due to IRs present but not reported, the minimum score was lowered to 30 which corresponds to repeats as small as 15 characters. Running the sequence, masking the IRs and rerunning eliminated the hit imbalance. These results suggest that indel seeds should be used in conjunction with hit filtering to avoid processing redundant clumped hits.

The IRs found in the dog sequence with arm separation > 500 000 bp (97% of those found) have not, to our knowledge, been reported before. Several examples found uniquely by the indel seed and the spaced seed are shown in Figure 2. The ones found by the indel seed are typical in that they tend to have higher indel to mismatch ratios than those tested in section 6.

8 CONCLUSION

We have presented a new seed model for homology search which explicitly allows indels. We give a waiting time formula for calculating indel seed sensitivity and show that indel seeds are superior to spaced and contiguous seeds in reasonable homology models which include indels. We discuss how indel seeds are used in practice and present the results of a limited search for inverted repeats in the dog genome using indel and spaced seeds with equivalent random hit rates.

REFERENCES

Aki, S., Kuboki, H. and Hirano, K. (1984). On discrete distributions of order *k*. *Ann. Inst. Statist. Math.*, **36**, 431–440.

Altschul, S., Gish, W., Miller, W., Myers, E. and Lipman, D. (1990). Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul, S., Madden, T., Schäffer, A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, **25**, 3389–3402.

Benson, G. (1999). Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Research*, **27**, 573–580.

Benson, G. (2005). TRDB at http://tandem.bu.edu/cgi-bin/trdb/trdb.exe.

Brejova, B., Brown, D. and Vinar, T. (2004). Optimal spaced seeds for homologous coding regions. *J Bioinform Comput Biol.*, **1**, 595–610.

Brejova, B., D.Brown, G. and Vinar, T. (2005). Vector seeds: an extension to spaced seeds. *Journal of Computer and System Sciences*, **70**(3), 364–380.

Buhler, J. and Sun, Y. (2005). Designing seeds for similarity search in genomic DNA. *Journal of Computing and System Sciences*, **70**, 342–363.

Choi, P.K. and Zhang, L. (2004). Sensitivity analysis and efficient method for identifying optimal spaced seeds. *Journal of Computer and System Sciences*, **68**, 22–40.

Choi, K., Zeng, F. and Zhang, L. (2004). Good spaced seeds for homology search. *Bioinformatics*, **20**, 1053–1059.

Keich, U., Li, M., Ma, B. and Tromp, J. (2004). On spaced seeds for similarity search. *Discrete Applied Mathematics*, **138**(3), 253–263.

Kent, W. (2002) BLAT–the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.

Lindblad-Toh, K. and *et al*. (2005). Genome sequence, comparative analysis and haplotype structure of the domestic dog. *Nature*, 438, 803–819.

Ma, B., Tromp, J. and Li, M. (2002). Patternhunter: faster and more sensitive homology search. *Bioinformatics*, **18**, 440–445.

Noe, L. and Kucherov, G. (2004). Improved hit criteria for dna local alignment. *BMC Bioinformatics*, **5**, 149–158.

Noe, L. and Kucherov, G. (2005). Yass: enhancing the sensitivity of DNA similarity search. *Nucleic Acids Research*, **33**, W540–W543.

Pearson, W. and Lipman, D. (1988). Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, **85**, 2444–2448.

Sun, Y. and Buhler, J. (2004). Designing multiple simultaneous seeds for dna similarity search. In *Proceedings, RECOMB*, pages 76–84.

Warburton, P., Giordano, J., Cheung, F., Gelfand, Y. and Benson, G. (2004). Inverted repeat structure of the human genome: the X chromosome contains a preponderance of large highly homologous inverted repeats which contain testes genes. *Genome Res.*, pages 1861–1869.

Xu, J., Brown, D., Li, M. and Ma, B. (2004). Optimizing multiple spaced seeds for homology search. In *Proceedings, Combinatorial Pattern Matching*.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e350–e358
doi:10.1093/bioinformatics/btl240

Efficient identification of DNA hybridization partners in a sequence database

Tobias P. Mann[1,*] and William Stafford Noble[1,2]

[1]Department of Genome Sciences and [2]Department of Computer Science and Engineering, University of Washington, Seattle WA, USA

ABSTRACT

Motivation: The specific hybridization of complementary DNA molecules underlies many widely used molecular biology assays, including the polymerase chain reaction and various types of microarray analysis. In order for such an assay to work well, the primer or probe must bind to its intended target, without also binding to additional sequences in the reaction mixture. For any given probe or primer, potential non-specific binding partners can be identified using state-of-the-art models of DNA binding stability. Unfortunately, these models rely on dynamic programming algorithms that are too slow to apply on a genomic scale.

Results: We present an algorithm that efficiently scans a DNA database for short (approximately 20–30 base) sequences that will bind to a query sequence. We use a filtering approach, in which a series of increasingly stringent filters is applied to a set of candidate *k*-mers. The *k*-mers that pass all filters are then located in the sequence database using a precomputed index, and an accurate model of DNA binding stability is applied to the sequence surrounding each of the *k*-mer occurrences. This approach reduces the time to identify all binding partners for a given DNA sequence in human genomic DNA by approximately three orders of magnitude, from two days for the ENCODE regions to less than one minute for typical queries. Our approach is scalable to large DNA sequences. Our method can scan the human genome for medium strength binding sites to a candidate PCR primer in an average of 34.5 minutes.

Availability: Software implementing the algorithms described here is available at http://noble.gs.washington.edu/proj/dna-binding

Contact: mann@gs.washington.edu

1 INTRODUCTION

Many fundamental methods in molecular biology rely on binding between complementary DNA molecules. For instance, the polymerase chain reaction (PCR) (Saiki *et al*., 1988) relies on the specific binding of short DNA primer sequences to the DNA of interest. PCR is used in a multitude of contexts (Innis *et al*., 1999), from disease diagnosis (Kaltenboeck and Wang, 2005) to gene expression measurement (Wong and Medrano, 2005). DNA microarrays (Schena *et al*., 1995) also rely on the specific hybridization of array probes to DNA sequences in a mixture in order to measure gene expression or determine sample genotypes (Stoughton, 2005).

Assays that rely on hybridization are compromised when primers or probes bind non-specifically to DNA molecules that are not their targets (Chou *et al*., 1992). In the presence of non-specific hybridization, measurement accuracy in quantitative assays can be severely compromised, especially when the hybridization target is present in low abundance. Even in the context of non-quantitative PCRs, non-specific binding can lead to the formation of undesired products that compete with the reaction of interest and reduce reaction yields. Therefore, assessing hybridization specificity is an important part of the design of these reactions.

The most straightforward approach to assessing hybridization specificity would be to query every potential binding site in the background DNA for binding affinity. In most experiments, the background DNA that comprises the reaction mixture consists of the genome of the organism being studied. Hence, for the human genome, this approach requires evaluating approximately six billion possible binding sites, corresponding to the two strands of each chromosome.

In practice, applying state-of-the-art DNA binding models on a genomic scale is not computationally feasible. These models use dynamic programming algorithms with a computational complexity of $O(nm)$ for two sequences of length m and n, respectively (Garel and Orland, 2004; Dimitrov and Zuker, 2004), and the complexity of querying an entire genome is $O(gmn)$, where g is the number of bases in the genome, m is the sequence length, and n is the size of the genomic subsequence queried at each position. In our experiments, scanning the complete human genome for binding sites to a 25-mer probe requires approximately 180 days of CPU time. For most primer or probe design applications, this is clearly too long to wait.

Current practical methods for predicting non-specific binding of a given DNA sequence rely on heuristic approximations. Perhaps the most commonly used method for identifying binding sites between a query DNA sequence and a target genome predicts binding sites based upon a pre-specified maximum number of mismatches between the probe's reverse complement and the target (Kent *et al*., 2002; Lowe *et al*., 1990; Wang and Seed, 2003; Xu *et al*., 2004). As we demonstrate below, this approach is inaccurate because sequences can stably bind in the presence of bulge loops, which correspond to insertions and deletions in an alignment. An alternative method for identifying non-specific binding sites relies on the BLAST algorithm or other alignment based criteria (Altschul *et al*., 1990; Haas *et al*., 2003; Zakour *et al*., 2004; Andersson *et al*., 2005). This approach, too, is inaccurate, primarily because BLAST is designed to detect statistically significant sequence homology, rather than sequence binding partners.

We propose a filter- and index-based method, shown in Figure 1, for rapidly identifying binding partners of a given query sequence. In

*To whom correspondence should be addressed.

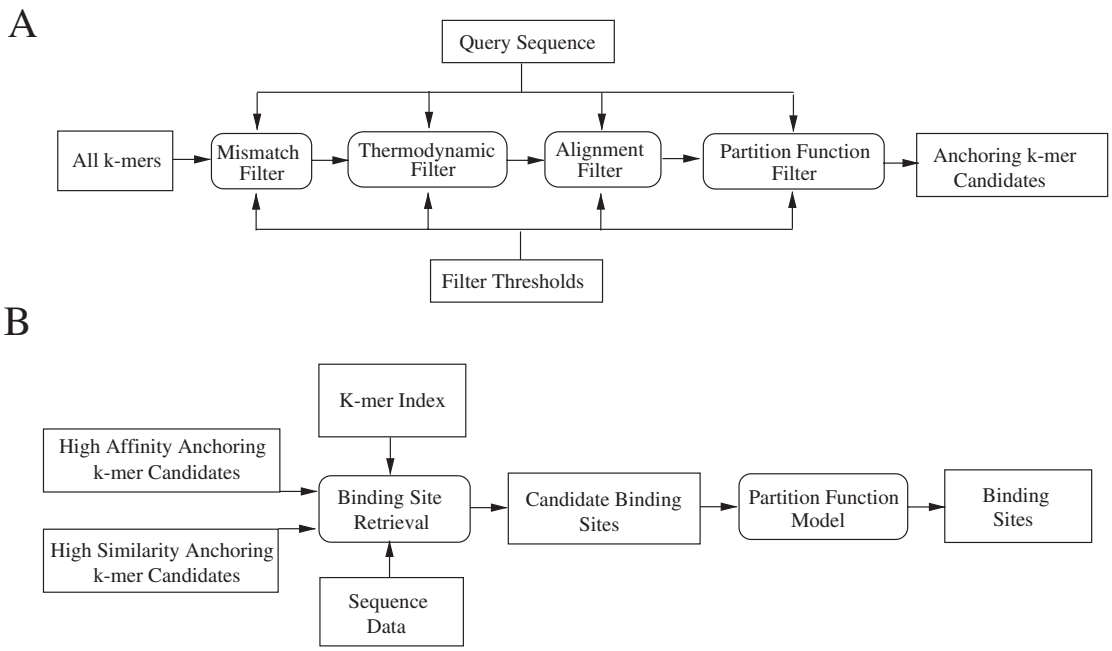


Fig. 1. Overview of filtering algorithm. **(A)** *k*-mer filtering. All *k*-mers for a specified value of *k* are input to the mismatch filter, along with a set of pre-chosen similarity thresholds. The four filters eliminate *k*-mers in turn, producing as output a list of candidate *k*-mers that could anchor a binding site. We subject all *k*-mers to two sets of thresholds, producing two sets of candidates binding site anchors. One set yields *k*-mers that have high thermodynamic affinity to the query, and the other set yields *k*-mers that have high sequence similarity to the query. (**B**) Candidate retrieval and evaluation. The *k*-mers that passed the filtering steps in (**A**) are located in the genome sequence using a precomputed index. We examine only those sites where a candidate *k*-mer from one group occurs with close proximity to a candidate *k*-mer from the other group. These candidate binding sites are then tested for binding affinity using the partition function model, and all sequences that bind to the query with greater than a target affinity are reported.

the initial stage (A), we consider all possible *k*-mers of a given length and identify *k*-mers that could anchor a binding site to the query sequence. This stage includes four filters that are designed to recognize various aspects of DNA binding stability. Two of the filters were developed for this application. The filters are applied in order of increasing computational complexity, so that most *k*-mers are excluded by the simplest filters. Using our approach and considering 10-mer anchors, typically more than 99% of target 10-mers are excluded from further consideration. In stage (B), we use a suffix array index to rapidly extract the sequence context of all occurrences of the *k*-mers obtained in the first step. These candidate binding sites are then evaluated using a model of DNA binding. Because so many *k*-mers are excluded at the outset, we can afford to apply an accurate model of DNA binding in the second stage of the algorithm.

Using our method, we achieve rapid and comprehensive identification of likely binding sequences. The first stage of the algorithm reduces the sequence search space by three orders of magnitude. The second stage is quick because many of the occurrences of the *k*-mers that pass the filtering stage can be eliminated by further filtering. Furthermore, our filter thresholds are set to achieve this speedup while retaining 100% accuracy, compared with considering every possible binding site in the target genome. Our approach reduces the amount of time to scan a sample 30 MB sequence from two days to under a minute for typical queries.

2 ALGORITHMS

We hypothesize that binding sites in genomic DNA can be comprehensively retrieved by first identifying short regions of agreement between the query sequence and the genomic DNA, and then examining the sequences containing these short regions of agreement with accurate models of DNA binding. We base this hypothesis on the observation that the thermodynamic instability of unbound bases in a DNA duplex (so-called 'loops') limits the amount of disagreement between a query sequence and any of its binding sites.

In particular, our method relies on a set of filters to identify *k*-mers that have good agreement with the query sequence, and could therefore anchor a binding site. In this section, we describe state-of-the-art models of DNA binding and then explain how our filters relate to those methods.

2.1 Partition function models of DNA binding

The overall goal of a model of DNA binding is to predict the *binding affinity* of a given pair of DNA sequences. The binding of two single stranded DNA molecules to form a dimer is a reversible reaction, and the binding affinity reflects the balance of association and dissociation reactions in a large population of molecules at thermodynamic equilibrium. When the binding affinity is large, then the dimer form is favored, and when the binding affinity is small, then the single stranded forms are favored. Currently, the most accurate models use thermodynamic reference data to approximate a quantity called the partition function. The partition function accounts for all ways in which two sequences can interact, and weights each interaction according to the energy of the interaction. The value of this function is proportional to the binding affinity.

In order to predict the binding affinity of two DNA sequences, partition function models of DNA binding stability consider physically realistic alignments between the two molecules, weighting each alignment according to its energy. The energy of a given alignment depends on a number of factors. The primary factor is the number of bases that are paired, and whether or not the paired bases are adjacent. In general, adjacent base pairs have higher binding energy than isolated base pairs due to so-called stacking interactions between adjacent base pairs. Conversely, runs of consecutive mismatches between the two strands, called loops, reduce the energy of the alignment. Extra energetic penalties are assigned to asymmetric loops. Bulge loops, corresponding to insertion and deletions in alignments, are also energetically unfavorable. Finally, the energetic stability of a single internal mismatch has been found to vary significantly according to the sequence context (SantaLucia, Jr, 1998; SantaLucia, Jr and Hicks, 2004), and these effects must also be taken into account.

Recently, efficient dynamic programming methods have been developed to compute the affinity of two DNA molecules (Garel and Orland, 2004; Dimitrov and Zuker, 2004). In this approach, a dynamic programming algorithm computes the sum of the exponentials of the energies of almost every alignment in which one molecule has at least one base pair with the other molecule. This sum is then proportional to the binding affinity. In this work, we use the HYBRID software (Markham and Zuker, 2004), which implements one such dynamic programming algorithm. However, our method does not rely on the specifics of the HYBRID software: our filters are designed to account for known, generic features of DNA affinity, and other models of DNA binding could be used in the final step to evaluate the filtered list of candidates. Indeed, although HYBRID and similar methods represent the state of the art in determining the affinity of two DNA sequences, they are known to systematically neglect some alignments that are important in some contexts.

2.2 An efficient algorithm for finding binding sites

Our goal is to identify all of the sequences in a database that bind to a query sequence according to a given partition function model of DNA binding. We do this in two stages, as described in Figure 1. First, we identify two groups of k-mers. One group of k-mers consists of k-mers with high sequence similarity to the query, and the other group of k-mers consists of k-mers with high thermodynamic affinity to the query sequence. Each group is defined as the set of k-mers that pass through a series of four filters described below; both groups are passed through the same filters but each group is identified by the use of different filter thresholds for each filter. In the second stage, each location in the sequence where there is a k-mer from the high affinity group within a pre-specified distance of a k-mer from the high similarity group is retrieved, along with flanking sequence. These candidate binding sites are then evaluated using the partition function model. The output of the algorithm is a list of binding partners for the query sequence.

In the first stage of our approach, we consider all k-mers of a given length, and we use a series of four filters to eliminate k-mers that have little affinity or similarity to the query. Each filter is designed to reject those k-mers that have little affinity to the query, and thus restrict the number of candidate binding sequences that must be considered. Furthermore, the filters are designed to be increasingly stringent, and are applied in order of increasing computational complexity; the first filter is very fast but will pass some k-mers with low affinity to the query, whereas the last filter is more expensive to compute but will reject all those k-mers with little thermodynamic affinity to the query. Each filter must be applied in conjunction with a threshold. The threshold for each filter is determined empirically by examining characteristics of binding sites predicted by the partition function model of DNA binding. These thresholds are chosen conservatively, so that each filter will pass some k-mers with low affinity to the query rather than discard k-mers that could anchor a binding site.

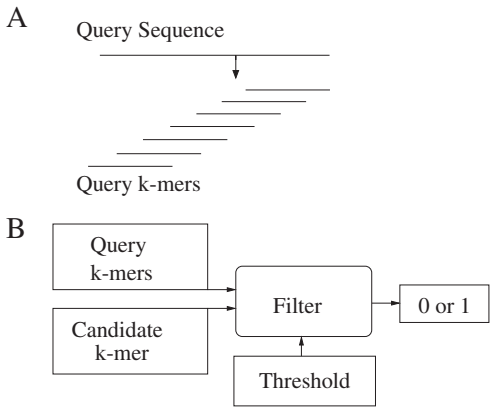


Fig. 2. Filtering k-mers. **(A)** Decomposition of the Query sequence into k-mers. The query sequence is decomposed into overlapping k-mers of a specified length. **(B)** Computation of the similarity of a k-mer to the query. Each filter identifies k-mers that could anchor a binding site, taking as input the k-mers derived from the query sequence, a candidate k-mer, and a pre-specified filter threshold. Each filter then reports whether the candidate k-mer had the specified level of similarity to at least one of the k-mers in the query sequence or not.

Each filter uses a function designed to compare two k-mers. In order to compare a candidate k-mer to a query sequence, we first decompose the query sequence into k-mers of the same length as the candidate k-mer, and then compare the candidate k-mer to each k-mer derived from the query (see Figure 2A). If any of the query derived k-mers meet the specified similarity to the candidate k-mer (Figure 2B), then the candidate k-mer is retained for further analysis. If none of the query derived k-mers meets the specified similarity, then the candidate k-mer is eliminated from further consideration.

The simplest filter—the mismatch filter—eliminates k-mers that differ from every k-mer in the query sequence by more than a specified number of bases. This filter is designed to reject k-mers that have little affinity to any part of the query sequence. The filter function computes the fraction of mismatches between a candidate k-mer K and the query sequence Q:

$$F_1(K,Q) = \max_{j \in s(Q,k)} \sum_{i=1}^{k} \frac{\delta(K_i = j_i)}{k},$$

where $s(Q,k)$ returns the set of all k-mers in Q, and δ is the Kronecker delta function.

The second filter rejects k-mers that contain destabilizing internal mismatches relative to the query. These destabilizing mismatches are identified using thermodynamic data on DNA binding stability (SantaLucia, Jr and Hicks, 2004). This filter's function is similar to

the mismatch filter, except that it takes into consideration the specific stabilities of dinucleotide stacks (pairs of adjacent, paired bases) and single internal mismatches. We implement this filter by encoding each k-mer K as a complex valued vector $\Phi(K)$, and we developed this filter so that the inner product of the conjugate of the encoding of one k-mer and another k-mer approximates the sum of the free energy of binding between the first k-mer and the reverse complement of the second k-mer, and vice versa. Details of this encoding are given in the appendix. The final value of this filter is a normalized dot product:

$$F_2(K,Q) = \max_{j \in s(Q,k)} \frac{\langle \Phi(K), \Phi(j) \rangle}{\sqrt{\langle \Phi(K), \Phi(K) \rangle \langle \Phi(j), \Phi(j) \rangle}}.$$

The third filter rejects k-mers that do not have good sequence agreement with the query, considering the possibility of asymmetric internal loops. For each candidate k-mer, this filter's function considers many alignments with respect to the query sequence, weighting each by the number of matches and the length and topology of loops. Asymmetric internal loops serve to separate regions of sequence agreement, and thus this filter will recognize sequence similarity even when regions of sequence agreement are separated by insertions or deletions in one sequence with respect to the other. We developed this filter function to be a coarse approximation of the partition function for one sequence binding to the reverse complement of the other, and we therefore consider only base pairing (and neglect the detailed thermodynamic reference data on dinucleotide stability) and internal loops of length three or less. In addition, we use loop stability values optimized for this application. The final value of the alignment filter is

$$F_3(K,Q) = \max_{j \in s(Q,k)} \frac{f(K,j)}{\sqrt{f(K,K) \cdot f(j,j)}}.$$

The alignment function $f(\cdot,\cdot)$ is described in the appendix.

The fourth filter applies the partition function model directly. In this step, we compute the binding affinity between the reverse complement of the k-mer and the query sequence. In order to normalize out the binding properties of the query sequence, we divide this binding energy by the binding energy of the k-mer in the reverse complement of the query sequence with the highest affinity to the query sequence. The final value is

$$F_4(K,Q) = \frac{g(\hat{K},Q)}{\max_{j \in s(Q,k)} g(\hat{j},Q)},$$

where a carat denotes reverse complement, and $g(.,.)$ is the partition function model of DNA binding. In practice, this filter is the most stringent and the most computationally complex.

We apply the four filters twice, with two sets of filter thresholds, to get two sets of candidate anchoring k-mers. We use filter thresholds so that the high similarity group of k-mers will be similar with respect to filters F1 and F2, and the high affinity k-mers will be similar with respect to filters F3 and F4. We then locate all occurrences of both candidate sets in the sequence database, and further consider only those locations in the sequence database where there is a k-mer from the high affinity group close to a k-mer from the high similarity group (see Figure 3).

After the four filtering steps, we must efficiently locate all occurrences of the high affinity and high similarity k-mers within the given sequence database. This is accomplished by using a modified suffix array (Gusfield, 1997; Manber and Myers, 1993) to index the database. In a suffix array, pointers to suffixes of a sequence are sorted lexicographically; in our modified suffix array, the pointers are sorted based on comparison of only the first k positions of the suffix, where k is the length of the filtered k-mers. We also build a hash table on the suffix array itself, so that the positions in the suffix array corresponding to a query k-mer can be quickly located (with a computational complexity of $O(k)$ per k-mer lookup). We use this sequence index, consisting of the modified suffix array and the hash table into the suffix array, to rapidly identify all locations where a candidate k-mer from one group occurs close to a candidate k-mer from the other group. These occurrences, along with their flanking sequences, comprise the list of candidate binding sites.

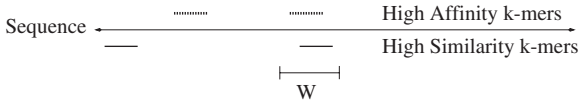


Fig. 3. Search for proximal hits. Our binding site search algorithm finds anchoring k-mers in the search sequence. We use two sets of filter thresholds, and obtain two sets of candidate anchoring k-mers; one set has high similarity to the query, and the other set has high affinity to the query (occurrences of k-mers from the high affinity set are drawn with dashes above the search sequence, and occurrences of k-mers with high similarity are drawn with solid lines below the search sequence). We locate all occurrences of both groups of candidate anchoring k-mers, and further examine only those sites where there is a candidate anchoring k-mer from the high similarity group occurring within a pre-specified distance w from a candidate anchoring k-mer from the high affinity group.

In the final step, each remaining candidate binding site is evaluated by the partition function model for affinity to the query sequence. As we show in Section 4, by using a set of fast, accurate filters, the filtering and indexing stages of the algorithm reduce the sequence search space by three to five orders of magnitude. Therefore, in the final step, we can afford to incorporate a relatively sophisticated, computationally expensive model of DNA stability. Thus, by coupling a pre-filtering step with accurate refinement of the candidate list, we achieve both efficiency and accuracy.

2.3 Choice of filter thresholds

Clearly, the success of our filtering strategy depends to a large extent on the thresholds that we use for each filter. If our thresholds are too stringent, then we risk eliminating true binding partners from our list. Conversely, if our thresholds are not stringent enough, then the efficiency of the search will decrease.

We compute these thresholds empirically by using the partition function model. First, with respect to a given set of experimental conditions and a target level of binding affinity, we scan a sequence database for binding sites to a set of query sequences using the partition function model, storing a list of all binding sites with stability better than a given threshold. We then choose filter parameters conservatively, so that if we re-searched the sequence using our filtering approach, we would obtain all of the binding sites obtained in the slow linear scan.

Our thresholds are set by analyzing the binding sites identified using a linear scan, using the procedure illustrated in Figure 4. We decompose each binding site into its constituent k-mers, as in

Table 1. Query sequences

	Query sequence	Length	GC	ΔG PCR	ΔG MA
1	GAGCTGCGGCAGAGGCTGGCGCCC	24	0.79	−24.5	−36.8
2	GCCTGCACTGGCTTCAGGAAGCTGGAGCC	29	0.65	−25.3	−40.1
3	GGCCAGTTCCTGCAGCCCGAGGC	23	0.74	−21.6	−33.2
4	AGTGGCATGCCTCTCTCTACCCAGC	25	0.60	−19.7	−32.2
5	CCACCAAAAAGTAATTAAAGGGTTTGCCTCAT	32	0.38	−19.5	−35.6
6	CACGCAAATCATCCCCAGCCACATC	25	0.56	−19.1	−31.8
7	CAGGTGTCCCTGCTTCGGCTTCCAG	25	0.64	−20.6	−33.3
8	CGCGAAGTGACCTTCAGAGAGTACGCCAT	29	0.55	−22.3	−37.2
9	CTGGACTGCCAAGTCCAGGGCAGGCC	26	0.69	−23.0	−36.1
10	GTCACCCACCTGCTGGCCCCGG	22	0.77	−20.9	−32.0
11	GGGGCTCAATAAGTCTGCTTCCACCTT	27	0.52	−19.5	−33.0
12	GGGTGAGGCCCATTCATAAGACTGGC	26	0.58	−19.6	−32.7
13	CCAGTCATGTTGCCCCGTTTGTCAGAG	27	0.56	−20.4	−34.1
14	GGGAGGGCTGAAGAGGGCACTCC	23	0.70	−19.4	−30.9
15	GGATGCATATGGACTCTTAGGTGTTCTGCG	30	0.50	−20.6	−36.0
16	GAAAGGGCTGGCTATGATAAACTGTGGC	28	0.50	−19.4	−33.7

ΔG PCR: Free energy of binding, in kilocalories per mole, of the sequence to its reverse complement at 55 C in 50 mM NaCl and 2 mM MgCl2; ΔG MA: Free energy of binding, in kilocalories per mole, of the sequence to its reverse complement at 40 C in 1 M NaCl. Energies are computed using the HYBRID software.

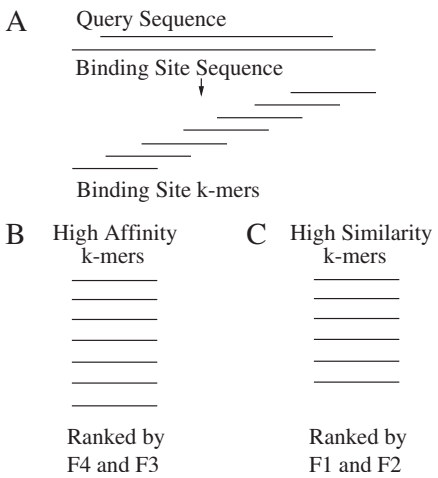


Fig. 4. Filter Analysis of a binding site. **(A)** Decomposition of binding site. Each binding site is decomposed into its constituent *k*-mers. **(B)** Ranking of binding site *k*-mers according to thermodynamic affinity. The binding site *k*-mers are ranked according to similarity to the query by F4; F3 is used to break ties. The similarity scores of the top ranked *k*-mer are added to the set of similarity scores used to determine filter thresholds for the high affinity group of candidate *k*-mer binding site anchors. **(C)** Ranking of binding site *k*-mers according to sequence similarity. All *k*-mers, except the top ranked *k*-mer in (**B**) are re-ranked according to sequence similarity to the query by F1; F2 is used to break ties. The similarity scores of the top ranked *k*-mer are added to the set of similarity scores used to determine filter thresholds for the high similarity group of candidate *k*-mer binding site anchors.

Figure 4(A). We then rank these *k*-mers according to similarity to the query sequence using the filter functions. For the high affinity group of filter parameters, we rank first by filter function F4 and break ties using filter function F3 as shown in Figure 4(B). For the high similarity group of filter parameters, we remove the top ranked *k*-mer and re-rank the remaining *k*-mers using the filter function F1; we break ties with filter function F2 as shown in Figure 4(C). We then compute the similarity of both top ranked *k*-mer to the query according to all filter functions.

After analyzing each binding site recovered from the linear scans, we integrate information from all binding sites as follows. Each binding site contributes two sets of similarity scores, one set for the top *k*-mer ranked according to thermodynamic affinity and one set for the top *k*-mer ranked according to sequence similarity. We accumulate all sets of similarity scores into two sets. One set contains the similarity scores for all top ranked *k*-mers according to thermodynamic affinity, and the other set contains the similarity scores for all top ranked *k*-mers according to sequence similarity. To obtain the final filter parameters, we find the minimum similarity score in a set over all binding sites for each filter function. This is thus a conservative method for obtaining filter parameters, and ensures that if the sequence were re-searched with our filtering approach, we would recover all of the binding sites identified with the linear scan.

Intuitively, the two sets of filter thresholds capture different characteristics of DNA binding: sequence agreement and *k*-mer binding affinity. These two notions of similarity are not the same: consider a query sequence that consists of several A bases followed by several G bases. A *k*-mer consisting entirely of A bases would have perfect sequence agreement to the left part the the sequence, whereas a *k*-mer consisting of all G bases with two consecutive internal A bases would have poor sequence agreement, but the reverse complement of that *k*-mer would have much higher binding affinity to the query sequence than the sequence consisting of all A bases. Our double filtering approach accounts for both situations.

3 METHODS

For validation purposes, we focus on the ENCODE regions of the human genome ENCODE Project Consortium (2004). These 44 regions together

Table 2. Rejection rates for the four filters

Sequence	F1	F2	F3	F4	Remaining
1	99.4	74.3	3.8	0.6	1589
2	99.2	69.0	7.3	2.8	2431
3	99.4	63.8	6.4	10.0	1862
4	99.3	54.0	2.9	24.0	2333
5	99.1	63.0	1.1	48.0	1887
6	99.3	51.7	3.1	29.7	2282
7	99.3	63.8	5.1	2.5	2322
8	99.2	65.0	7.3	14.5	2414
9	99.3	66.5	6.9	10.0	2059
10	99.5	58.5	2.9	13.6	1969
11	99.3	59.8	6.5	26.6	2164
12	99.3	63.2	5.7	47.1	1360
13	99.3	63.7	3.9	20.2	2181
14	99.4	75.4	2.8	2.7	1419
15	99.1	67.9	8.0	12.1	2375
16	99.2	68.5	4.0	47.4	1316
mean	99.3	64.3	4.9	19.5	1998

The table lists, for each of the query sequences in Table 1, the percentage of *k*-mers rejected by each of the four filters using the high affinityfilter thresholds, as well as the total number of *k*-mers that pass through all four filters. These results are for weak binding sites in standard PCR conditions.

Table 3. *k*-mer filtering performance

Sequence	PCR		Microarray	
	Weak	Medium	Weak	Medium
1	1589	15	15	15
2	2431	20	20	20
3	1862	14	14	14
4	2333	16	16	16
5	1887	19	20	16
6	2282	16	16	16
7	2322	16	16	16
8	2414	20	20	20
9	2059	17	17	17
10	1969	13	13	13
11	2164	18	18	18
12	1360	15	17	14
13	2181	18	18	18
14	1419	14	14	14
15	2375	21	21	21
16	1316	16	17	13
mean	1997.7	16.8	17.0	16.3

The table lists, for each of the query sequences in Table 1, the total number of *k*-mers that pass through all four filters using the high affinity thresholds.

Table 4. Proximity filtering performance

Sequence	PCR		Microarray	
	Weak	Medium	Weak	Medium
1	69	88	66	94
2	55	71	50	57
3	59	77	71	83
4	69	80	70	94
5	76	93	91	92
6	57	67	63	74
7	57	48	48	64
8	80	98	81	98
9	54	70	62	82
10	58	72	59	91
11	61	65	64	73
12	63	74	76	85
13	68	94	80	97
14	48	69	56	89
15	74	89	79	95
16	63	85	79	88
mean	63.19	77.50	68.44	84.75

The table lists, for each of the query sequences in Table 1, the percentage of sequence locations that are rejected by the proximity filtering step. The final row contains the column average.

comprise 1% of the human genome. The ENCODE regions were chosen to be representative of the entire genome, based on gene density, GC content, and density of conserved non-coding elements.

In addition, we chose a collection of sixteen query sequences to use in our experiments. We manually selected from the ENCODE regions exonic and intronic sequences that vary in length from 22 to 31 bases. Each selected sequence was analyzed using HYBRID, assuming standard PCR conditions (see below). A selected sequence *S* was added to the query set if the binding affinity between *S* and its reverse complement is greater than -19 kilocalories per mole. None of the selected query sequences overlaps a repeat sequence as annotated by RepeatMasker, and the percent GC of the queries range from 40% to 80%. The final list of query sequences is given in Table 1.

To generate a gold standard set of binding sites, we used HYBRID to scan every base of both strands of the ENCODE regions. The scan employed a window size of 35 bases, and was repeated with two different sets of experimental conditions, typical of PCRs and microarray experiments, respectively. For PCR conditions, we predict binding affinities at 55 C, with a concentration of 50 millimolar NaCl and 2 millimolar $MgCl_2$. For microarray conditions, we predict binding affinities at 40 C, with a concentration of 1 molar NaCl. In subsequent experiments, we used these lists of binding sites to verify that our algorithm correctly identifies all binding sites.

In selecting filter thresholds, we focus on two levels of binding site stringency, corresponding to weak and medium binding. We define a weak binding site as one where the equilibrium constant of the dimer formed by the binding site and the query sequence is at most six orders of magnitude less than the dimer formed by the query sequence binding to its reverse complement, under equal initial single strand concentrations. We define medium binding sites similarly, except we require only three orders of magnitude of difference. We used all binding sites recovered with the linear scans to choose filter thresholds.

4 RESULTS

In order to measure the efficiency and accuracy of our binding site prediction algorithm, we scan the ENCODE regions with a collection of query sequences, using HYBRID with and without the filtering and indexing pipeline. This experiment shows that our approach yields a significant improvement in running time, without missing any binding sites.

We begin by examining the behavior of each of the four filters for the thresholds designed to detect *k*-mers with high thermodynamic affinity to the query. Table 2 lists the percent of *k*-mers eliminated by the combined filters for each of the 16 query sequences. The mismatch kernel appears to provide the most value, since it has a

Table 5. Number of candidate sequences examined and accepted by the partition function model of DNA binding, and time for each run

Sequence	Weak PCR			Medium PCR			Weak microarray			Medium microarray		
	Candidates	Actual	Time	Candidates	Actual	Time	Candidates	Actual	Time	Candidates	Actual	Time
1	30712	25	6 m	2340	19	18 s	9543	21	35 s	1332	16	23 s
2	57587	23	20 m	11994	15	40 s	18702	16	76 s	3326	11	19 s
3	44628	100	8 m	4030	20	21 s	4882	21	17 s	3078	17	19 s
4	35218	45	11 m	5269	19	22 s	6152	20	24 s	1178	16	12 s
5	23235	29	6 m	1870	13	18 s	2791	14	23 s	1132	9	12 s
6	91220	108	13 m	7304	19	26 s	8112	21	28 s	6301	16	21 s
7	33780	48	10 m	6667	20	31 s	6667	22	24 s	4962	15	28 s
8	22310	26	5 m	179	14	10 s	3025	15	19 s	99	10	12 s
9	35396	45	12 m	6552	18	22 s	8390	19	35 s	4264	14	17 s
10	175109	336	12 m	5741	21	17 s	7976	25	29 s	1909	18	11 s
11	75547	40	16 m	5908	17	24 s	6181	18	32 s	4896	14	28 s
12	20887	70	6 m	3120	18	16 s	3369	20	19 s	1598	14	14 s
13	22934	31	6 m	907	19	14 s	3276	20	16 s	418	14	13 s
14	142717	361	13 m	5221	21	22 s	7081	37	34 s	1837	17	13 s
15	20106	26	8 m	1413	14	13 s	2839	16	17 s	153	10	12 s
16	17138	29	6 m	2988	17	17 s	3680	19	19 s	1639	13	17 s
mean	53032.8	83.9	9.9 m	4468.9	17.8	20.7 s	6416.6	20.3	27.9 s	2382.6	14.0	16.9 s

The table lists, for each experiment, the total number of candidate sites produced by the filtering and indexing pipeline, the number of those sites that are considered by HYBRID to be true binding sites, and the total wall clock time required to identify the sites.

rejection rate over 99%; however, this high rejection rate is primarily a result of its placement first in the filter pipeline. In practice, the more computationally expensive filters are also more exclusive. In each case, the filters reduce the complete set of $4^{10} = 1,048,576$ k-mers to less than 2500 k-mers. Also, note that the setup in Table 2 (weak binding sites in PCR conditions) is the most permissive and hence yields a relatively large number of k-mers. Table 3 lists the total number of k-mers that successfully pass through all four filters in each experiment: strong and weak binding, and PCR and microarray conditions. With the exception of the weak binding/PCR conditions, the algorithm typically produces on the order of 20 k-mers for further consideration. The results in Tables 2 and 3 use the filter thresholds selected using the high affinity filter parameters. Results for the high similarity set of thresholds are similar.

After obtaining both groups of candidate binding site anchors, we then identify locations in the sequence where a k-mer from the high affinity group occurs near a k-mer from the high similarity group. Table 4 lists, for all four experiments, the percentage of sites identified by the high affinity group of binding site anchor candidates that are not close enough to a k-mer occurrence from the high similarity group of binding site anchor candidates. On average, this step reduces the list of candidate sites by between 63% and 85%, depending upon the experiment.

The final stage of the analysis involves running HYBRID on the filtered list of candidate binding sites. Table 5 lists, for each experiment, the number of candidate binding sites that were evaluated by the HYBRID software. Clearly, this stage is very important, since the number of sites considered is typically several orders of magnitude larger than the number of sites that HYBRID identifies as binding partners. In this sense, our filters are conservative: they do not very closely approximate the computation performed by HYBRID. However, these conservative thresholds lead to high accuracy. For all 16 primers that we tested, our filtering and indexing pipeline identifies 100% of the binding sites that were identified by HYBRID in the much more computationally expensive linear scan of the entire ENCODE regions. Furthermore, as shown in Table 5, the entire pipeline is very efficient. For medium binding strength and standard PCR conditions, HYBRID was only required to evaluate an average of 4467 sites, and scanning the entire ENCODE database required 20.7 seconds on average. By comparison, a linear scan of the ENCODE regions using HYBRID takes approximately two days.

5 DISCUSSION

We have presented a method for rapidly identifying binding partners for a given query DNA sequence within a genome-sized DNA database. Our approach combines a k-mer filtering method, which identifies k-mers that could nucleate binding sites to the query, with an efficient indexing method, which rapidly locates these nucleating k-mers in a sequence database. The combination of these two methods speeds up the DNA binding site search by at least three orders of magnitude.

We note that not all predicted binding sites will be relevant to every hybridization reaction. Some dimers may be slow to reach equilibrium concentrations, especially if the dimer has internal loops. Thus, in a PCR, some dimers may not have time to form and thus may not be a problem. However, in microarray hybridization experiments, conditions are much closer to equilibrium, and secondary binding sites may be more of a concern.

Among the four tasks that we considered, finding weak binding partners for PCR primers is the most difficult search task, and the one for which we obtain the least improvement. However, this task may be the most important for experimentalists, because even weak binding sites can drive high yields on undesired background reactions. This is because in PCR, the primers are present in vast excess,

and the excess concentration of primer in the initial stages of the reaction drives high levels of weak binding site occupancy, even though the binding affinity is low.

The major bottleneck in our method is evaluating the final list of sequences. Even though we reduce the number of sequences that must be considered by several orders of magnitude, the partition function model is still sufficiently slow that it introduces a significant computational burden. It is important to recognize, however, that we can typically place an upper limit on this burden: once we identify a pre-specified number of binding partners for a given query, the search can terminate, since that particular query is not a tenable primer or probe candidate.

6 FUTURE WORK

Conceptually, searching a RNA database for binding sites to a RNA sequence is similar to the problem addressed in this paper. Although the same partition function model can be used to compute the binding affinity of one RNA molecule for another, the parameters are different due to the chemical differences between RNA and DNA Mathews *et al*. (1999). We are currently beginning experiments to evaluate the computational complexity of this version of the binding site search problem. Further, it may also be of interest to search for DNA binding partners of an RNA molecule, or RNA binding partners for a DNA molecule. Because the data for these heterogeneous dimers is much less complete than the data for DNA/DNA or RNA/RNA dimers, our method is not applicable to these binding site searches.

Our method depends critically on the filter parameters, and clearly the similarity of the anchoring k-mers in a binding site to a query is not known in advance. We are therefore increasing the size of our database of predicted binding sites, so that we can estimate the sensitivity of our method for a wider variety of query sequences.

7 CONCLUSIONS

We have shown that DNA binding site search of genomic scale DNA sequences is tractable for realistic experimental conditions, for primer length DNA sequences. Our filters work together to reduce by at least three orders of magnitude the number of sequences that must be examined by a partition function model of DNA binding, reducing search time from two days to scan the ENCODE regions to under a minute for typical queries. This filter- and index-based method will be useful in the design of PCR primers and short oligonucleotide probes.

ACKNOWLEDGEMENTS

This work was funded by NIH awards R33 HG003070, T32 HG00035 and R01 GM071923.

REFERENCES

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. A basic local alignment search tool. 215: 403–410, 1990.

A. Andersson, R. Bernander, and P. Nilsson. Dual-genome primer design for construction of DNA microarrays. *Bioinformatics*, 21(3): 325–332, 2005.

Q. Chou, M. Russell, D. E. Birch, J. Raymond, and W. Bloch. Prevention of pre-PCR mis-priming and primer dimerization improves low-copy-number amplifications. 20(7):1717 –1723, 1992.

R. A. Dimitrov and M. Zuker. Prediction of hybridization and melting for double stranded nucleic acids. *Biophys. Journal*, 87(1): 215–226,2004 .

ENCODE Project Consortium. The ENCODE (ENcyclopedia of DNA Elements) project. *Science*, 306: 636–640, 2004.

T. Garel and H. Orland. Generalized Poland-Scheraga model for DNA hybridization. *Biopolymers*, 75(6): 453–467, 2004.

D. Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, Cambridge, UK, 1997.

S. A. Haas, M. Hild, A. P. H. Wright, T. Hain, D. Talibi, and M. Vingron. Genome-scale design of PCR primers and long oligomers for DNA microarrays. *Nucleic Acids Res*., 31(19): 5576–5581, 2003.

M. A. Innis, D. H. Gelfand, and J. J. Sninsky. *PCR Applications: Protocols for Functional Genomics*. Academic Press, 1999.

B. Kaltenboeck and C.M. Wang. Advances in real-time PCR: Application to clinical laboratory diagnostics. *Adv. in Clin. Chem*., 40:219–259, 2005.

W.J. Kent, C. W. Sugnet, T.S. Furey, K.M. Roskin, T.H. Pringle, A.M. Zahler, and D. Haussler. The human genome browser at UCSC. *Genome Res*., 12(6): 996–1006, 2002.

T. Lowe, J. Sharefkin, S. Q. Yang, and C. W. Dieffenbach. A computer program for selection of oligonucleotide primers for polymerase chain reactions. *Nucleic Acids Res*., 18(7):1757 –1761, 1990.

U. Manber and E. Myers. Suffix arrays:a new method for on-line search. *SIAM J. Comput*., 2: 935–948, 1993.

N. Markham and M. Zuker. DINAMelt web server for nucleic acid melting prediction. *Nucleic Acid Res*., 33: W577–W581, 2004.

D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol*., 288: 911–940, 1999.

R. K. Saiki, D. H. Gelfand, S. Stoffel, S.J. Scharf, R. Higuchi, G. T. Horn, K. B. Mullis, and H.A.Erlich. Primer-directed enzymatic amplification of DNA with a thermostable polymerase. *Science*, 239(4839): 487–491, 1988.

J. SantaLucia, Jr. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc. Natl Acad. Sci. USA*, 95: 1460–1465, 1998.

J. SantaLucia, Jr and D. Hicks. The thermodynamics of DNA structural motifs. *Annu. Rev. Biophys. Biomol. Struct*., 33: 415–440, 2004.

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. 270: 467–470, 1995.

R. B. Stoughton. Applications of DNA microarrays in biology. *Annual Rev. Biochem*., 74: 53–82, 2005.

X. Wang and B. Seed. A PCR primer bank for quantitative gene expression analysis. *Nucleic Acids Res*., 31(24): e154, 2003.

M.L. Wong and J.F. Medrano. Real-time PCR for mRNA quantitation. *Biotech*., 39: 75–85, 2005.

W. Xu, W. J. Briggs, J. Padolina, W. Liu, C. R. Linder, and D. P. Miranker. Using MoBioS' scalable genome join to find conserved primer pair candidates between two genomes. *Bioinformatics*, 20: i355–i362, 2004.

N. Ben Zakour, M. Gautier, R. Andonov, D. Lavenier, P. Veber M-F. Cochet, A. Sorokin, and Y. Le Loir. GenoFrag: software to design primers optimized for whole genome scanning by long-range PCR amplification. *Nucleic Acids Res*., 32(1): 17–24, 2004.

APPENDIX: FILTERS

We use four filters. The simplest counts the number of mismatches between two k-mers, and the most complicated computes the binding energy of the reverse complement of a k-mer binding to the query according to the partition function model. The other two filters are described in the next two subsections. We use A and B to represent the sequences input to the filter; these sequences have length m and n, respectively. We use A_i to represent the ith element of sequence A.

Free energy filter

The free energy filter is defined first by mapping sequences A and B to complex valued vectors $\Phi(A)$ and $\Phi(B)$, and then taking their inner product. We developed the mapping Φ and present it here for the first time.

The mapping function has the property that if A and B are identical, then

$$\langle \Phi(A)^*, \Phi(B) \rangle = \Delta G(A, \hat{B}) + \Delta G(B, \hat{A}) \text{-} \Delta G_i$$

where a carat denotes reverse complement, and $\Delta G(A, \hat{B})$ is the free energy of binding of A to the reverse complement of B, and ΔG_i is a duplex initiation energy parameter. This computation of the binding energy between two sequences approximates the free energy computations presented in SantaLucia, Jr and Hicks(2004).

The inner product $\langle \Phi(A), \Phi(B) \rangle$ has the property that the angle between $\Phi(A)$ and $\Phi(B)$ increases with the number of mismatches. The angle is also sensitive to the identity of the mismatching bases, and will increase more for strongly destabilizing mismatches (such as C—C) than for mildly destabilizing mismatches (such as G—G).

The inner product can be computed as

$$\langle \Phi(A), \Phi(B) \rangle = \sum_{k=1}^{m-1} [\Delta G_s(A_k, A_{k+1}) \Delta G_s(B_k, B_{k+1})] + [\delta(A_k = B_k) \delta(A_{k+1} = B_{k+1})]$$

where $\Delta G_s(A_k, A_{k+1})$is the free energy of binding of the dinucleotide stack(SantaLucia, Jr and Hicks, 2004).

Alignment filter

We designed the alignment filter to coarsely approximate the partition function model of DNA binding. This filter computes a score that rewards runs of consecutive identical bases in each sequence, and that penalizes loops analogously to the loop entropy functions in(SantaLucia, Jr and Hicks, 2004). The parameters that we use to reward consecutive matches and penalize loops were optimized for this application.

Table 6. The loop penalty matrix

1.050	0.120	0.010
0.120	0.800	0.003
0.010	0.003	0.003

The filter value is computed first by filling a dynamic programming matrix, and then computing the sum of all of its entries. This filter uses an AT reward parameter α, and a GC reward parameter β. We set $\alpha = 1.1$ and $\beta = 1.15$. This is analogous to assigning a slightly more stable energy to GC base pairs than AT base pairs, but this filter neglects specific dinucleotide effects.

The dynamic programming matrix is filled in as follows. If A_i is not equal to B_j, then $F_{i,j}$ is set to zero. Otherwise, if A_i is equal to B_j, then

$$F_{i,j} = \max_{i-3 \le x < i,\, j-3 \le y < j} (R * F_{x,y} * L[i-x, j-y])$$

where $R = \alpha$ if A_i and B_j are both A or T, and $R = \beta$ otherwise. The loop penalty matrix L is given in Table 6. The element in the first row and column is greater than 1 in order to reward consecutive matches.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e359–e367
doi:10.1093/bioinformatics/btl217

An experimental metagenome data management and analysis system

Victor M. Markowitz[1,*], Natalia Ivanova[2], Krishna Palaniappan[1], Ernest Szeto[1], Frank Korzeniewski[1], Athanasios Lykidis[2], Iain Anderson[2], Konstantinos Mavrommatis[2], Victor Kunin[3], Hector Garcia Martin[3], Inna Dubchak[2], Phil Hugenholtz[3] and Nikos C. Kyrpides[2]

[1]Biological Data Management and Technology Center, Lawrence Berkeley National Lab, USA, [2]Genome Biology Program, Joint Genome Institute, USA and [3]Microbial Ecology Program, Joint Genome Institute, USA

ABSTRACT

The application of shotgun sequencing to environmental samples has revealed a new universe of microbial community genomes (metagenomes) involving previously uncultured organisms. Metagenome analysis, which is expected to provide a comprehensive picture of the gene functions and metabolic capacity for microbial communities, needs to be conducted in the context of a comprehensive data management and analysis system. We present in this paper IMG/M, an experimental metagenome data management and analysis system that is based on the Integrated Microbial Genomes (IMG) system. IMG/M provides tools and viewers for analyzing both metagenomes and isolate genomes individually or in a comparative context. IMG/M is available at http://img.jgi.doe.gov/m.

Contact: vmmarkowitz@lbl.gov

1 INTRODUCTION

Environmental microbial community (microbiome) genome analysis, also known as *metagenome* analysis, (Riesenfeld *et al*., 2004) is expected to lead to advances in environmental cleanup, agriculture, industrial processes, and alternative energy production. Similarly, human metagenome analysis could provide new insights into the variation of microbial populations associated with the human body, ascribe qualitative and quantitative changes in human microbiota as risk/causative factors of disease and lead to the development of new treatment strategies (Gordon *et al*., 2005).

The application of shotgun sequencing to microbiome samples has enabled the study of metagenomes involving previously uncultured and unculturable organisms. Comparative analysis of the metagenomes in the context of available reference isolate genomes could potentially reveal large-scale patterns of biochemical interactions and habitat-specific correlations in the host environment that might otherwise be missed (DeLong and Karl, 2005). Studies of environmental microbiomes, such as acid mine drainage biofilms (Tyson *et al*., 2004) and Sargasso Sea samples (Venter *et al*., 2004), as well as studies of human microbiomes, such as the human gut microbiome (Gordon *et al*., 2005), are examples of a rapidly expanding area of metagenome analysis applications.

Unlike microbial genome data from isolate organisms, the generation and interpretation of metagenome data is in early stages of development. Metagenomes sequenced by organizations such as the Joint Genome Institute (JGI), TIGR, and the Venter Institute, follow an assembly and annotation process that is specific to each sequencing center. Although traditional assembly and annotation algorithms do not perform as well on metagenome sequences as they do on isolate microbial genomes (see (Chen and Pachter, 2005) for an overview of metagenome sequence assembly and gene prediction problems), they yield data that are amenable to valuable comparative analysis and interpretation as illustrated by the studies published in (Tringe *et al*., 2005) and (Tyson *et al*., 2004). Thus, the metagenome sequences of simple microbiomes can be assembled into sizable scaffolds and for highly abundant (dominant) member organisms the quality of the assembly and annotation may approach that of draft isolate genomes. For such metagenomes, it is possible to infer the metabolic capabilities of dominant organisms and identify the key member organisms that perform community-essential tasks.

Although metagenome sequence data processing poses numerous challenges due to the complex nature and inherent incompleteness of the data, and the lack of methods designed specifically for processing such data, successful analysis can be carried out on existing metagenomic data. As initial methods are improved or new methods emerge, metagenome data sets will be revised, thus leading to better quality data and annotations. However, metagenome data analysis needs to be conducted in the context of a comprehensive data management and analysis system that provides support for data review and revision. We have addressed this need by developing an experimental metagenome data management and analysis system, IMG/M, based on the Integrated Microbial Genomes (IMG) system (Markowitz *et al*., 2006).

Like IMG, IMG/M is based on the principle that integration of available genomic data is essential for understanding the biology of newly sequenced genomes, as the efficiency of genome analysis increases substantially when it is conducted in a comparative

*To whom correspondence should be addressed.

context. Such an integrated context is even more critical for analyzing the inherently incomplete metagenome data. IMG/M has been successfully used for the study of biological phosphorus removing (EBPR) sludge communities (Martin *et al*., 2006), and is currently used for analyzing several metagenomes sequenced at JGI.

In the following sections, we first discuss the main metagenome data processing challenges. Next, we briefly review metagenome data modeling and analysis. Finally, we present the IMG/M metagenome data analysis tools and discuss our plans to extend these tools.

2 METAGENOME DATA PROCESSING

There are two general sequencing strategies to obtain genome sequence data from microbiome samples: directed sequencing and shotgun sequencing of random clones. Directed sequencing is either (i) function-driven, whereby clone libraries from a microbiome sample are sequenced after being screened for a desired function; or (ii) driven by phylogenetic markers, whereby the DNA flanking taxonomic anchors, such as 16S rDNA, is sequenced in large-insert libraries. Conversely, shotgun sequencing of microbiome sample clone libraries follows a relatively unbiased approach, which provides a broad survey of the gene content and metabolic capabilities of a microbiome. A combination of shotgun and directed sequence approaches may emerge in the future and thus combine the advantages of the broad coverage provided by shotgun sequencing with the ability of sampling specific genome areas in low abundance organisms without over-sequencing more abundant members of the microbiome. The discussion below pertains to metagenome data generated using shotgun sequencing.

Metagenome sequence data processing follows assembly and annotation procedures that are specific to each sequencing center. Assemblers, such as the Celera Genome Assembler, PHRAP, and JAZZ (Aparicio *et al*., 2002) have been used with mixed results (Chen and Pachter, 2005). Assembly of shotgun-sequenced microbiome samples poses a serious challenge to traditional assembly methods, due to a fundamental difference between the sequences derived from cultivated microbes and microbial communities. While the genome sequence of a cultivated microbe is derived from a clonal isolate, where all cells are descendants of one cell and therefore genetically identical or nearly identical, the aggregated genome sequence of a microbiome is derived from a heterogeneous pool of cells, some of which are genetically related and probably correspond to different strains of the same species, while others are genetically distinct. Although co-assembly of the sequences derived from different species does not seem to be a problem, traditional methods are not consistent in assembling the sequence reads belonging to different strains of the same species: depending on the assembly algorithm and sequencing read depth they can be resolved into strain-specific scaffolds or co-assembled into a composite species population scaffold. In the later case the strain-specific variations appear as single nucleotide polymorphisms (SNPs) in the sequence.

Annotation of the assembled metagenomes is also currently carried out using traditional approaches developed for isolate genomes. For instance, protein-coding genes (CDSs) are predicted on scaffolds and/or so called *shrapnel* sequences (single reads that are not incorporated into scaffolds) using microbial gene finders, such as Glimmer (Delcher *et al*., 1999) or Fgenesb (Soft-Berry, 2006). Performance of traditional gene prediction methods is affected by the inevitable fragmentation of metagenomic sequences, which in turn leads to fragmentation of the genes, and therefore sometimes gene prediction is limited to BLASTx of all open reading frames against protein sequence databases. Functional annotation of predicted CDSs is generally carried out using COG (Tatusov *et al*., 1997), Pfam (Bateman *et al*., 2004), InterPro (Mulder *et al*., 2005), and KEGG (Kanehisa *et al*., 2004); functional annotations can also be marred by gene fragmentation in the metagenome datasets.

Sometimes an additional stage of scaffold *binning* is included in order to assign scaffolds and *shrapnel* sequences to organism types (phylotypes) that could range from coarse-level groupings such as domain (*Bacteria*, *Archaea*) down to fine-level groupings such as individual strains of a given species. It is highly desirable that all sequence fragments are assigned to a particular strain in the community; however, this is usually not feasible due to the different abundance of the strains and variation of sequence coverage. Consequently, the highest resolution grouping for metagenome data can be achieved at the species level, that is, grouping together genomic fragments that are likely derived from members of a given *species population*, whereby each bin represents a snapshot of a *composite genome* of a *species population*. Some regions of such a *composite genome* are represented by sequences originating from only one strain (usually, the most abundant one), while others are covered by sequences from multiple strains. The latter may exhibit different types of strain-level heterogeneity, from SNPs to extensive genome rearrangements. Binning algorithms rely on measuring the oligonucleotide frequency in different scaffolds, depth of sequence coverage or phylogeny of conserved protein markers; thus, binning accuracy depends on the sequence coverage, quality of the assembly, scaffold size, complexity of the microbiome, and available reference isolate microbial genomes (Chen and Pachter, 2005). While it is expected that binning will be difficult in the case of highly fragmented metagenomes of complex microbiomes, such as those from soil samples (Tringe *et al*., 2005), for simpler microbiomes with sufficient sequence coverage it is possible to reconstruct more than 95% of the individual genomes of the dominant community members (Tyson *et al*., 2004).

Despite the metagenome data processing challenges mentioned above, analysis of metagenomes does not need to wait for the development of optimal data generation and annotation methods: such analysis can be carried out with existing methods with the results of these analyses serving as a basis for improving the methods in an iterative process.

3 METAGENOME DATA MODEL AND ANALYSIS

Similar to isolate microbial genome data, *metagenome* data captures information about DNA sequences along with *genes* that can be further characterized in terms of *functional roles*. A *gene* represents an ordered sequence of nucleotides located on a particular *chromosome* that encodes a specific product (i.e., a protein or RNA molecule); its protein product can be characterized in terms of sequence similarity to other protein products, presence or absence of *conserved motifs* and *domains*. *Functional* roles of genes can be characterized in the context of *pathways*, whereby pathways are associated with genes via gene products that can function as

enzymes catalyzing individual reactions of metabolic pathways. Similar to isolate microbes, the metabolic capacity of a whole *microbiome* can be characterized by analyzing the *metabolic maps* inferred from the gene content and distribution of its composite genome.

Metagenome data have an additional level of complexity reflecting the complex nature of microbiomes, which, unlike clonal isolates, consist of heterogeneous pools of cells belonging to different strains and species. Therefore metagenome scaffolds can be further characterized in terms of their *bin* assignment, whereby a *bin* could correspond to a *composite genome* of a *species population* or another higher-level taxonomic group. If a *bin* corresponds to the *species population*, it could be characterized by strain-level heterogeneity (e.g., SNPs or genome rearrangments). Similar to a metagenome which represents a random sample of the aggregate microbiome genome, a *bin* may represent only a subset of the aggregate genome of a *species population*, and therefore may not reflect all the diversity of this species population in terms of strain-level heterogeneity.

Another important difference between metagenome data and isolate genome data is that metagenome data are representative of a microbiome in a specific *host* environment and a specific *sample* of this environment. Sample (meta) data characterizing the biological material collected for sequencing, are specific to an application domain. For example, for biomedical applications samples are collected from human donors and therefore are associated with attributes that describe donor host data (e.g., demographic and clinical record), sample structural and morphological characteristics (e. g., site and time of collection) and sample processing protocol. Sample metadata are critical in metagenome comparative data analysis.

Comparative data analysis plays an important role in understanding the biology of isolate microbial genomes (Bowers *et al*., 2004). Similar to isolate genomes, the analysis of metagenomes in the comparative context of other (e.g., phylogenetically related) genomes is substantially more efficient than analyzing each metagenome in isolation. Metagenome data analysis is set in a multidimensional data space, whereby microbiome samples form one of the dimensions and are analyzed in the context of other dimensions, such as component species populations, gene families represented by homolog/ortholog clusters, COG groups or Pfam families, and pathways and networks.

For example, microbiome samples can be compared in terms of presence and abundance of certain gene families. This type of analysis is based on the assumption that the genes important for adaptation to a particular environment will be found in many (if not all) organisms in the microbiome; moreover, such genes might be present in multiple copies, therefore, they are more likely to be found among the abundant gene families. Gene family abundance profiles can be analyzed at higher resolution, when bins within the same microbiome rather than microbiome samples are compared; this type of analysis allows to verify directly the assumption that abundant gene families are indeed present in many members of a microbiome.

Another emerging method of analyzing metagenomic data involves detection of presence and abundance of certain metabolic pathways in a specific microbiome sample or across samples of the same microbiome or different microbiomes. Such analysis typically involves examining *occurrence profiles* (Osterman and Overbeek, 2003) of functions and pathways of interest across samples associated with a specific microbiome or across diverse microbiomes. Alternatively, the bins within the same metagenome dataset can be compared in terms of presence/abundance of functions and pathways. This analysis helps to infer the metabolic capabilities of the component organisms in the community, and thus identify the key members of the microbiome that perform community-essential tasks and pinpoint the metabolic interactions within the microbiome and between the microbiome and its host environment.

Both examples discussed above are focused on the analysis of metagenome data *per se*, however, an efficient analysis of metagenomes is not possible without the context of reference genomes. Similar to comparisons of microbiome samples and bins within metagenome datasets, metagenome sequences can be compared to isolate microbial genomes in terms of gene family abundance, presence or absence of functions and pathways, and so on.

4 AN EXPERIMENTAL METAGENOME DATA MANAGEMENT AND ANALYSIS SYSTEM

We have developed an experimental metagenome data management and analysis system, IMG/M, based on the Integrated Microbial Genomes (IMG) system (Markowitz *et al*., 2006). The IMG/M system and data analysis tools are briefly overviewed below.

4.1 System Overview

The content of IMG/M can be seen as a superset of IMG's content. IMG integrates bacterial, archaeal and selected eukaryotic genomic data collected from multiple data sources. Thus, IMG 1.3 (as of December 1st, 2005) contains a total of 678 genomes consisting of 377 bacterial, 26 archaeal, 15 eukaryotic genomes and 260 bacterial phages. IMG's extensive collection of microbial genomes (both draft and finished) provides the foundation for analyzing the fragmented inventory of genes, functions, and organisms in microbiomes and their component populations.

In addition to the isolate genomes in IMG 1.3, the first experimental version of IMG/M (as of March 1st, 2006) includes metagenome sequences generated from an acid mine drainage (AMD) biofilm (Tyson *et al*., 2004), an agricultural soil sample (Tringe *et al*., 2005), three isolated deep sea ''whale fall'' carcasses (Smith and Baco, 2003), and two biological phosphorus removing (EBPR) sludge samples (Martin *et al*., 2006). These microbiomes comprise a representative set in terms of species diversity, abundance of dominant organism(s) and sequencing depth. For instance, species diversity ranges from very low in the case of the AMD sample to extremely high in the soil sample, while abundance of dominant organism(s) ranges from less than 1% in the soil sample to more than 80% in EBPR sludge samples. Furthermore, two EBPR sludge samples represent an example of microbiomes inhabiting similar environments in two distinct geographical locations. Consequently, the metagenome data in IMG/M can be employed to test use case scenarios, formulate and test various hypothesis, assess performance of available tools and develop new methods for metagenome analysis.

The IMG/M back-end consists of a data warehouse, sequence databases for similarity (BLAST) searches, and various auxiliary data files containing scaffold DNA sequences, pathway map images, and cached data for improving performance, such as pre-computed statistics and homolog results. An additional

BLAST database supports similarity searches based on the sequencing reads for analysis of strain level single nucleotide polymorphisms (SNPs). The data generated by microbial genome and metagenome data processing pipelines serve as input for a custom ETL (Extract-Transform-Load) toolkit that loads data into the IMG/M data warehouse. This toolkit is also employed for extracting, cleaning, integrating, and loading additional genomic and contextual data from external resources into the data warehouse. Additional custom tools are employed to compute gene relationships and clusters and load these data into the data warehouse.

The data model for the IMG/M data warehouse allows integrating primary genomic sequence information, computationally predicted and curated gene models, pre-computed sequence similarity relationships, and functional annotations and pathway information in a coherent biological context. Isolate organisms are identified via their taxonomic lineage (domain, phylum, class, order, family, genus, species, strain). For each genome, the primary DNA sequence and its organization in scaffolds and/or contigs, are recorded. Genomic features, such as predicted coding sequences (CDSs) and some functional RNAs, are also recorded. Protein-coding genes are further characterized in terms of molecular function and participation in pathways. Proteins are grouped into protein families based on sequence similarity. Pathways, reactions, and compounds are included from KEGG and LIGAND. Additional functional annotations according to Gene Ontology terms (Gene Ontology Consortium, 2004) are provided by EBI Genome Reviews (Kersey *et al*., 2005), while COG provides clusters of orthologous groups of genes. Ortholog and paralog gene relationships for isolate microbial organisms are computed based on bidirectional best hit (BBH) with clusters formed using Markov Clustering method (MCL) (Enright *et al*., 2002). Isolate organisms are characterized in terms of phenotypes (e.g., morphology, geochemistry), ecotype (including geographical coordinates) and disease.

Microbiome samples are treated as ''meta'' organisms with the collection of their associated genes forming their respective metagenomes. The sequences of a microbiome sample together with their associated genes and annotations are organized in bins when possible, with multiple bins providing support for recording data generated using different binning methods. Similar to isolate organisms, microbiome samples are characterized in terms of phenotypes, ecotype, disease, and relevance. These data are only minimal in coverage, reflecting the current scarcity of such data for microbiome samples.

4.2 Data Analysis

We review below the IMG/M data exploration and comparative analysis tools, with special emphasis on the support for metagenome analysis. IMG/M tools can be also employed for analyzing isolate microbial genomes in the same way as their IMG counterparts.

4.2.1 Data Exploration Data exploration tools in IMG/M help selecting and examining genomes, genes, and functions of interest. Metagenomes as well as isolate genomes can be selected using a keyword based *Genome Search* in conjunction with a number of filters or an alphabetically or phylogenetically organized *Genome Browser*. Microbiomes can be further examined using the *Microbiome Details*, where a user can find relevant metadata, such as geographical location, along with various summaries of interest, such as the total number of scaffolds and genes or the number of genes associated with functional characterizations (eg., COG, Pfam), as shown in the right pane of Figure 1. *Microbiome Details* also provides an estimate of phylum level assignment (*Phylogenetic Mapping*) of metagenomic fragments in the sample based on sequence comparison to isolate genomes. This overview consists of the distribution of the best BLAST hits at different percent identity thresholds of a metaproteome (i.e., the collection of all the proteins encoded in the metagenome) of interest against the proteomes of all isolate genomes in the system, as shown in the left pane of Figure 1. For each metagenome one can also examine the associated list of scaffolds and contigs, and information on individual bins and their scaffolds when bins are available.

Genes can be selected using a keyword based *Gene Search*, sequence similarity search tools, or a gene profile based selection tool, the *Phylogenetic Profiler*, discussed in more detail below. The functional role of genes in IMG/M is characterized by a variety of annotations, including their COG membership, association with Pfam domains, Gene Ontology (GO) assignments, and association with enzymes in KEGG pathways. Functional annotations can be searched using keywords and filters, with the selected functions leading to a list of associated genes either directly or via a list of organisms. COG functional categories and KEGG pathways can be searched and browsed separately. The lists of genes and functional annotations that are of interest for further exploration can be maintained using various *Analysis Carts*, which are similar to shopping carts of commercial websites.

Individual genes can be analyzed using *Gene Details* pages, as illustrated in Figure 1. A Gene Information table includes gene identification, locus information, biochemical properties of the product, and associated KEGG pathways. *Gene Details* also includes evidence for the functional prediction: gene neighborhood, COG, InterPro, and Pfam, and pre-computed lists of homologs, orthologs and paralogs (for isolate organisms), or intra-metagenome homologs as well as homologs to other genomes and metagenomes (for microbiomes). The gene neighborhood displays the target gene and its homologs in user selected related genomes with its neighboring genes in a 25kb chromosomal window: for example, the gene neighborhood in the *Gene Details* in Figure 1 shows the target gene (centered, in red) and other genes within a 25kb window. The *Gene Neighborhoods* in Figure 1 shows the neighborhood of a target gene of the *Ferroplasma acidarmanus Type I* bin of the AMD metagenome, compared to homologous genes of the *Ferroplasma acidarmanus fer*1 isolate genome: each gene's neighborhood appears above and below a single line showing the genes reading in one direction on top and those reading in the opposite direction on the bottom; genes with the same color indicate association with the same COG. For each gene, locus tag, scaffold coordinates, and COG number are provided locally (by placing the cursor over the gene), while additional information is available in the *Gene Details* associated with each gene. A gene can be also examined in the context of its associated pathways, whereby the link embedded in the pathway name listed in the *Gene Information* table allows the KEGG map associated with the gene to be displayed. On such a map, EC numbers are color-coded and linked to the *Gene Details* for the associated genes.

Individual COG categories can be further explored with *COG Category Details* that lists the COGs of a given category and

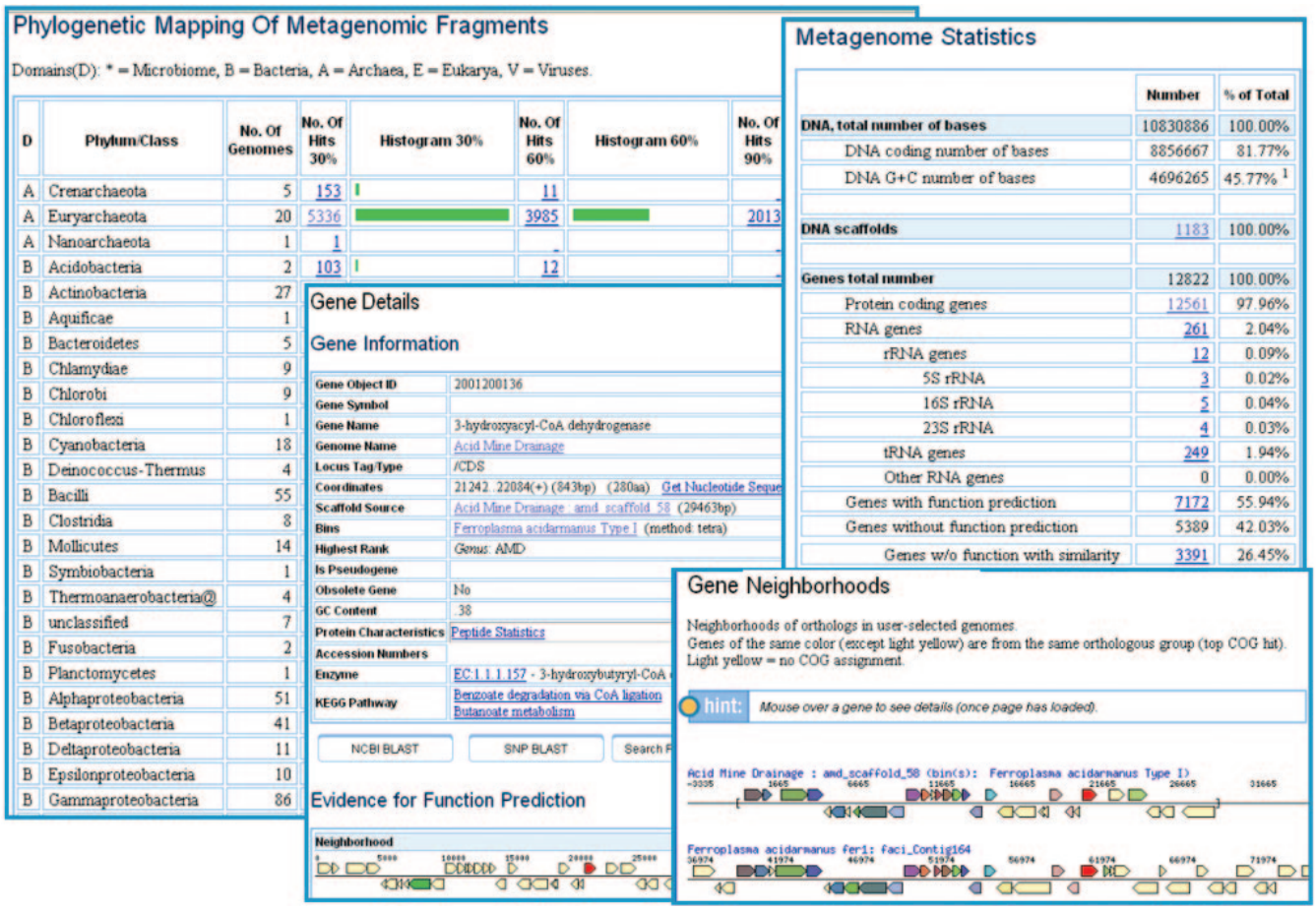


Fig. 1. AMD Microbiome Details: Metagenome Statistics and Phylogentic Mapping of Fragments. Gene Details and Gene Neighborhoods Example for an AMD Metagenome Gene.

the number of organisms that have genes belonging to each COG. For a given COG, the ''organism counts'' are linked to a list of organisms and their associated ''gene counts''. Gene counts for all COGs in a given category can be displayed for multiple organisms using *COG Profile*. KEGG pathways can be explored in a similar manner using *KEGG Pathway Details* and *Enzyme Profile*. *COG Profile* and *Enzyme Profile* are further discussed below.

4.2.2 Comparative Data Analysis The gene content of metagenomes and genomes can be examined with a profile-based selection tool, gene neighborhood analysis tools, and multiple sequence alignment tools. Functional annotations can be examined with several occurrence and abundance profile-based tools. We discuss below in more detail the profile based selection, occurrence profile, and abundance profile tools.

The *Phylogenetic Profiler* tool allows comparing the gene content of a target entity (microbiome, bin, or isolate organism) to that of other entities (microbiomes, bins or organisms) by defining a *profile* for the genes of the target entity in terms of presence or absence of homologs in other entities. Similarity cutoffs can be used to fine-tune the selection. Similar to isolate genomes, differences in gene content between metagenomes can be correlated with a specific phenotype or environment, while the comparison of the gene content of bins within the metagenome helps inferring the metabolic capabilities of the component populations and identify the organisms that may be responsible for community-essential tasks. The example shown in Figure 2 illustrates how the *Phylogenetic Profiler* helps finding differences in gene content between the component populations in the Acid Mine Drainage (AMD) microbiome. In this example, genes in the bin corresponding to *Leptospirillum* sp. group III that have no homologs in other bins in this metagenome are identified. Among the ''unique'' genes in *Leptospirillum* sp. group III one can find those responsible for nitrogen fixation, shown in the *Phylogenetic Profiler Results* pane of Figure 2, which makes this organism a keystone species in the AMD microbiome due to limitation of external nitrogen sources (Tyson *et al*., 2004).

Occurrence profile tools allow examining profiles of genes and functions across microbiomes, bins, and isolate organisms. Gene occurrence profiles usually involve genes within the same bin or organism: if such genes have similar occurrence profiles across other bins or organisms, then they may also have a similar evolutionary history and may potentially be functionally linked, or co-regulated in a pathway (Bowers *et al*., 2004). The profile for a gene x, across bins or organisms y_1 to y_n has the form of a vector $(L_1, \ldots, L_n)$ where L_i represents a set of y_i genes that are

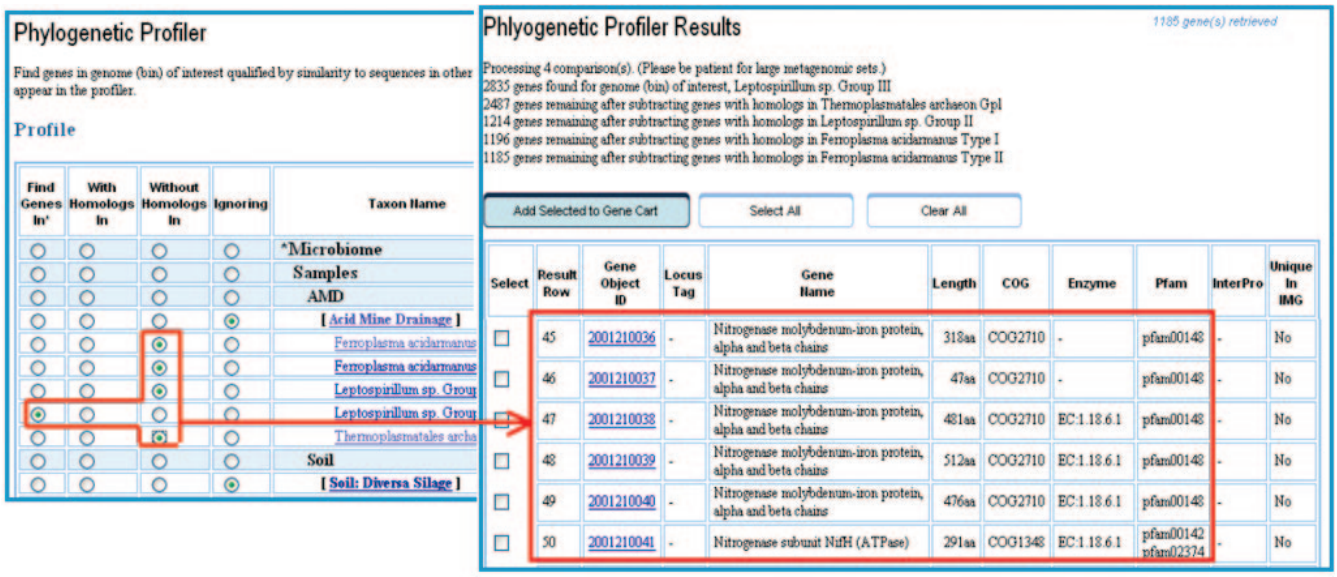


Fig. 2. Finding Gene Content Differences with the Phylogenetic Profiler Between AMD *Leptospirillum* sp. group III Bin and other AMD Bins.

associated with x, where the association of y_i genes with x is based on a specific sequence similarity method.

Functional occurrence profile tools, such as *COG Profile*, *Pfam Profile*, and *Enzyme Profile*, show the occurrence profiles for functional characterizations such as COGs, Pfam families, or enzymes involved in pathways across the selected entities (microbiomes, bins and organisms). Individual COGs, Pfam families, or enzymes are selected using a variety of search and browse tools and are maintained using *COG*, *Pfam*, and *Enzyme Carts*, respectively.

The occurrence profile for a specific function, f, shows the pattern of f across the selected entities, y_1 to y_n, in the form of a vector of the form $(L_1, \ldots, L_n)$, where L_i represents the set of y_i genes that are associated with f. Functional occurrence profiles provide an estimate of the similarity between entities in terms of association with a specific pathway or functional characterization.

The example shown in Figure 3 illustrates how occurrence profiles for a custom list of Pfam families can be used to predict the presence of a pathway for CO_2 fixation in metagenome data sets. The first step in one of CO_2 fixation pathways is catalyzed by anaerobic carbon monoxide dehydrogenase. A keyword search on expression ''CO dehydrogenase'' with Pfam as a filter (see *Search Terms and Pathways* pane of Figure 3) retrieves a list of six Pfam families, as shown in the *Function Search Results* pane of Figure 3. Four of these Pfam families correspond to different subunits of anaerobic carbon monoxide dehydrogenase, and therefore are selected and saved using the *Pfam Cart*. The occurrence profiles for these Pfam families are then computed and displayed in a tabular form as shown in the *Pfam Profile* pane of Figure 3, with each row displaying the profile of a specific Pfam across three whale-fall microbiomes and five bins of the AMD microbiome. Each cell in the profile result table contains a link to the associated list of genes and displays the count (*abundance*) of genes in the list. Colors are used to represent visually gene abundance, whereby white, bisque and yellow represent gene counts of 0, 1-4, and over 4 respectively. The occurrence profiles shown in Figure 3 indicate that, despite the presence of several spurious hits, anaerobic CO dehydrogenase is most likely absent from the organisms in the AMD microbiome and therefore these organisms probably rely on some other pathway of CO_2 fixation. Surprisingly, the genes coding for anaerobic CO dehydrogenase appear to be present in 2 out of 3 whale-fall microbiomes, as shown in in Figure 3. Occurrence profile tools provide two (functions vs. genomes, genomes vs. functions) display options for data visualization purposes.

An *Abundance Profile* tool allows comparing functional occurrence profiles for all COGs, Pfam families, or KEGG enzymes across microbiomes, bins, and isolate organisms of interest. This tool is especially useful for analysis of datasets obtained from the communities with high species diversity, where little or no sequence assembly can be achieved: for such datasets identification of predominant protein families allows users to infer habitat-specific biological traits.

The example in Figure 4 shows the abundance profiles of COGs displayed using a heat map, across the low-complexity AMD microbiome and the highly complex soil and whale-fall microbiomes. Arrows indicate COGs that are clearly overrepresented in the soil microbiome (bright red) as compared to other microbiomes (pink, orange, yellow and green); both COGs correspond to *glycosyl hydrolases* of different specificity. One would indeed expect to find glycosyl hydrolases abundant in microbiomes, such as those found in soil, that perform degradation of plant-derived carbohydrate polymers.

IMG/M also provides a tool for analysis of *strain-level heterogeneity* within a species population in metagenome data. *SNP BLAST* allows users to run BLASTn of nucleotide sequence of genes or scaffolds of interest in a metagenome, against a database of sequencing reads that were assembled to produce a composite species genome sequence comprised of multiple strains sequence types.

5 CONCLUSION

We have presented in this paper IMG/M, an experimental metagenome data management and analysis system. IMG/M provides support for the exploration and comparative analysis of

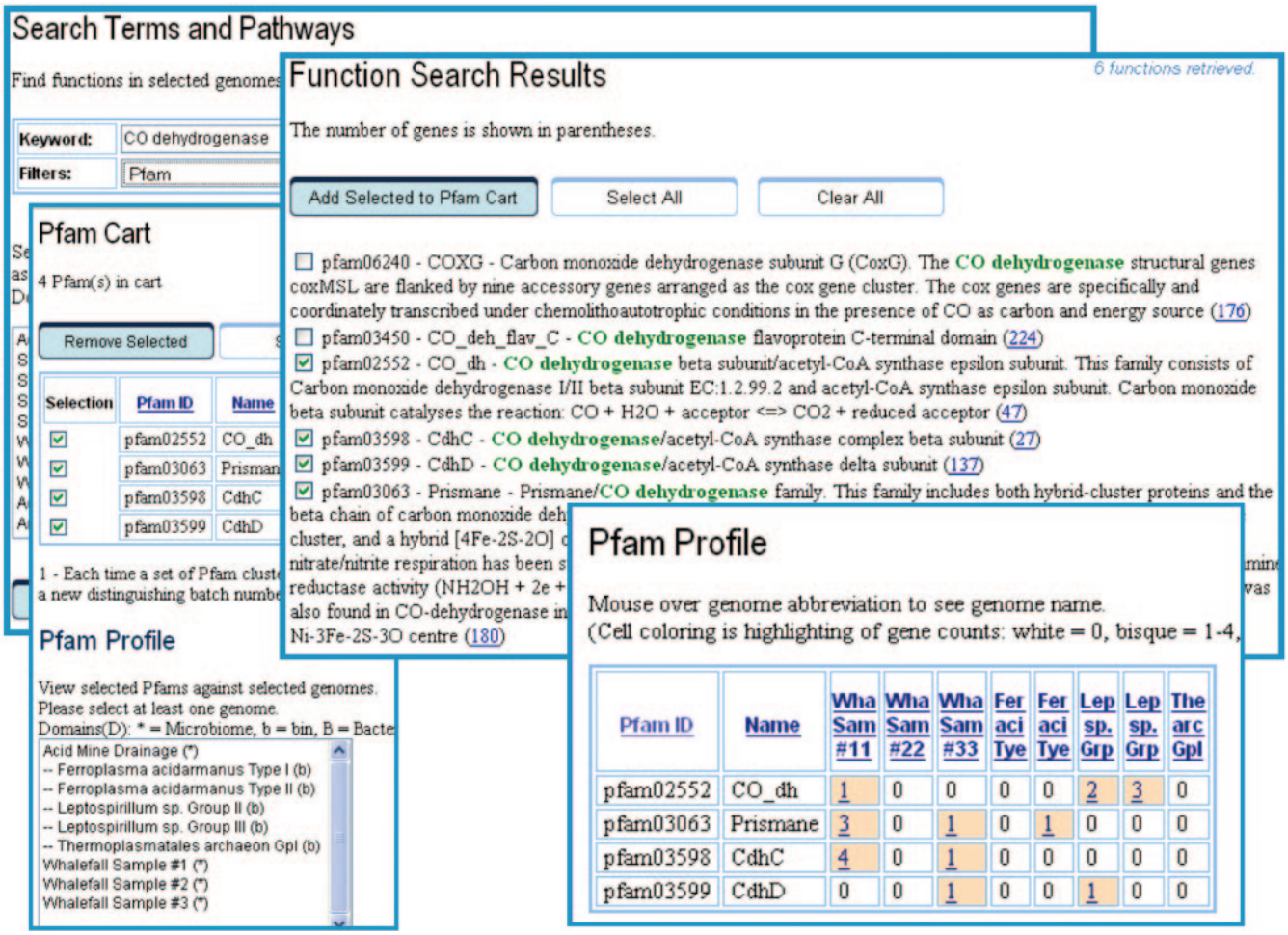


Fig. 3. Exploring the Presence of a Pathway for CO_2 Fixation Across Several Metagenomes with *Pfam Profile*.

metagenomes and their component populations in the context of other metagenomes and isolate genomes. IMG/M has been successfully used for the study of EBPR sludge communities (Martin *et al.*, 2006), and continues to be used for analyzing metagenomes sequenced at JGI, such as the *Olavius algarvensis* symbionts[1] and the termite gut microbial community[2]. Although IMG/M seems to be best suited for the analysis of low-complexity microbiomes, the system can be also used to infer the presence of important physiological characteristics in any microbiome and its species populations.

We plan to extend the tools provided by IMG/M in order to address several metagenome data analysis challenges. The first challenge regards the size and complexity of some metagenome data sets. Additional viewers need to be developed in order to improve the efficiency of analyzing such data sets via graphical representation of phenomena of interest set in a biological context.

A second challenge is posed by existing methods for binning metagenome scaffolds. These methods are in an early stage of development and have not been properly tested on metagenomes of complex microbiomes. We have found that some of these methods do not perform well even when applied to low diversity microbiomes in IMG/M, resulting in a significant number of unclassified or misclassified scaffolds. While analysis of environmental microbiomes is often function-driven and focuses on the genes and metabolic pathways of interest regardless of their assignment to a certain species, binning of scaffolds is essential for drawing a connection between the presence of certain genes (e.g. pathogenicity factors) and species composition of a microbiome. Consequently, there is an immediate need for tools that would provide support for comparing different binning methods and for assessing their accuracy, as well as for revising bins in terms of scaffold composition and gene content. We plan to develop tools for reviewing and curating the content of bins in IMG/M.

Finally, metagenome analysis tools need to be extended in order to account for the stochastic nature of metagenome data and variations in data quality due to incomplete sequence coverage. In most microbiomes a few dominant species tend to get the most sequencing coverage, sometimes approaching that of draft isolate genomes, while low abundance organisms can be represented by a small number of scaffolds or even single sequencing reads. Accordingly,

[1]http://www.jgi.doe.gov/sequencing/why/CSP2005/algarvensis.html
[2]http://www.jgi.doe.gov/sequencing/why/CSP2006/termitegut.html

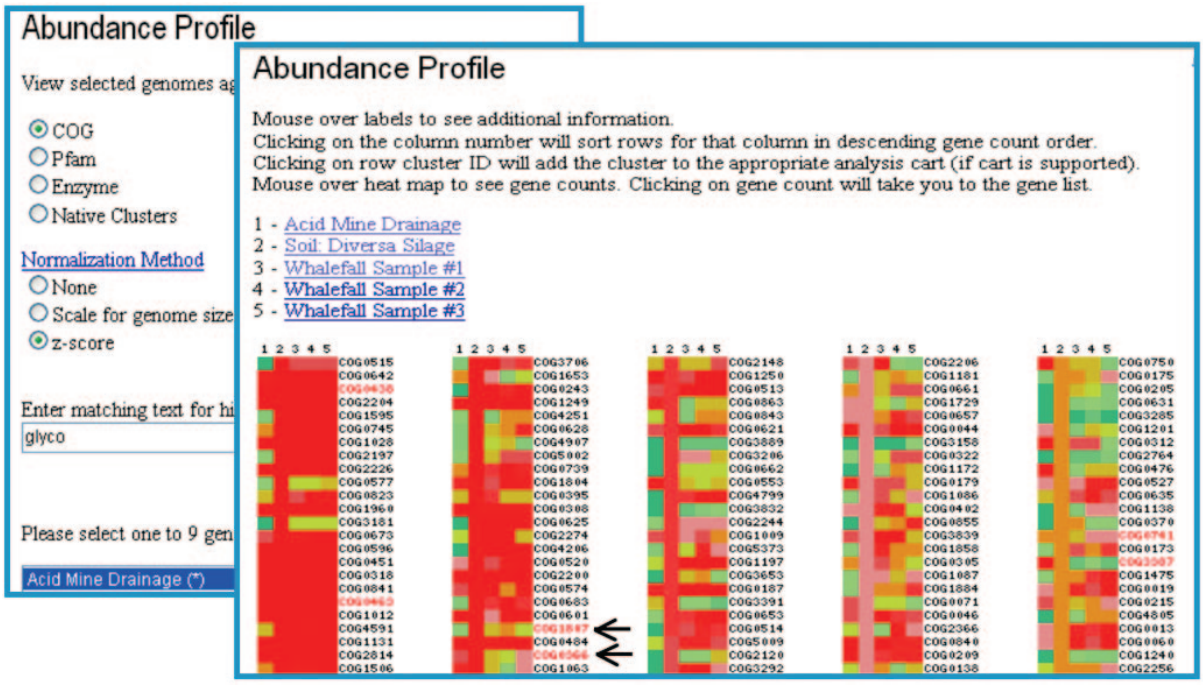


Fig. 4. Use of Abundance Profiles to Identify COG Families Overrepresented in the Soil Metagenome.

statistical tests need to be devised to estimate the sequence coverage of the bins and whether it is adequate for certain types of comparative analyses, such as metabolic reconstruction of pathways. Additionally, when metagenomes are compared to each other or to isolate genomes, statistical tests are needed for estimating the *statistical significance* of the observed differences. For example, the analysis of *Abundance Profiles* described above requires testing whether the differences in abundance can be ascribed to chance variation or not.

We also plan to extend the data model underlying the system in order to enhance its ability to capture *metadata* characterizing microbiome samples. Such metadata are often specific to an application (e.g., biomedical, ecological) domain. Samples are associated with properties used for metagenome analysis, such as sample structural and morphological characteristics (e.g. sample site, time of collection) and donor or host data (e.g. demographic and clinical record, including diagnosis, disease, stage of disease, and treatment information for human donors). Samples may also be involved in clinical studies and therefore can be grouped into several time/treatment study groups. In addition to extending the data model for supporting sample metadata, we plan to improve the coherence and completeness of these annotations via manual curation. In IMG/M, metadata such as disease, phenotype, ecotype and relevance for the isolate genomes were collected from sources such as GOLD (Liolios *et al*., 2006), while the microbiome sample metadata have been collected from published supplemental information and manually curated. The scarcity of metadata for isolate organisms and microbiome samples is a well known problem (Field and Hughes, 2005). We plan to collaborate with community standardization efforts in the metagenome data domain in order to ensure high coverage and consistence of microbiome sample metadata.

ACKNOWLEDGEMENTS

We thank all our colleagues who have contributed to the development of IMG and IMG/M. Special thanks to Eddy Rubin and James Bristow for their encouragement throughout this project. The work presented in this paper was supported by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

REFERENCES

Aparicio,S., Chapman,J., Stupka,E., Putnam,N., Chia,J. *et al*. (2002) Whole Genome Shotgun Assembly and Analysis of the Genome of Fugu rubripes. *Science*, **297**, 1301–1310.

Bateman,A., Coin,L., Durbin,R., Finn,R.D., Hollich,V. *et al*. (2004) The Pfam Protein Families Database. *Nucleic Acids Research*, **32**, D138–D141.

Bowers,P.M., Pellegrini,M., Thompson,M.J., Fierro,J., Yeates,T.O., Eisenberg,D. *et al*. (2004) Prolinks: A Database of Protein Functional Linkages Derived from Coevolution. *Genome Biology*, **5**.

Chen,K. and Pachter,L. (2005) Bioinformatics for Whole-Genome Shotgun Sequencing of Microbial Communities. *PLoS Computational Biology*, **1**(2), e24.

Delcher,A.L., Harmon,D., Kasif,S., White,O. and and Salzberg,S.L. (1999) Improved Microbial Gene Identification with Glimmer. *Nucleic Acids Research*, **27**(23), 4636–4641.

DeLong,E.F. and Karl,D.M. (2005) Genomic Perspectives in Microbial Oceanography. *Nature*, **437**, 336–342.

Enright,A.J., Van Dongen,S. and Ouzounis,C.A. (2002) An Efficient Algorithm for Large-Scale Detection of Protein Families. *Nucleic Acids Res*., **30(7)**, 1575–1584.

Field,D. and Hughes,J. (2005) Cataloguing our Current Genome Collection. *Microbiology*, **151(4)**, 1016–1019.

Gene Ontology Consortium. (2004), The Gene Ontology Database and Informatics Resource. *Nucleic Acids Research*, **32**, 258–261.

Gordon,J.I., Ley,R.E., Ruth,E., Ley, Wilson,R., Mardis,E., Xu,J., Fraser,C. and Relman,D.A. (2005) Extending Our View of Self: the Human Gut Microbiome Initiative (HGMI), http://www.genome.gov/Pages/Research/Sequencing/SeqProposals/HGMISeq.pdf

Kanehisa,M., Goto,S., Kawashima,S., Okuno,Y., Hattori,M. *et al*. (2004) The KEGG Resource for Deciphering the Genome. *Nucleic Acids Research*, **32**, D277–D280.

Kanz,C., Aldebert,P., Althorpe,N. *et al*. (2005) The EMBL Nucleotide Sequence Database. *Nucleic Acid Research*, **33**, D29–D33.

Kersey,P., Bower,L., Morris,L. *et al*. (2005) Integr8 and Genome Reviews: Integrated Views of Complete Genomes and Proteoms. *Nucleic Acid Research*, **33**, D297–D302.

Liolios,K., Tavernarakis,N., Hugenholtz,P., Kyrpides,N.C. *et al*. (2006) The Genomes On Line Database (GOLD) v.2: A Monitor of Genome Projects Worldwide. *Nucleic Acid Research*, **34**, D332–D334.

Markowitz,V.M., Korzeniewski,F., Palaniappan,K., Szeto,E. *et al*. (2006) The Integrated Microbial Genomes (IMG) System, *Nucleic Acids Research*, **34**, D344–D348.

Martin,H.G., Ivanova,I., Kunin,V., Warnecke,F. *et al*. Genetic Blueprints for Phosphorus Removal from Sludge Based on Metagenomic Sequencing. Submitted for publication. See http://www.jgi.doe.gov/sequencing/why/CSP2005/PO4accum.html.

Mulder,N.J., Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A. *et al*. (2005) InterPro, Progress and Status in 2005. *Nucleic Acids Research*, **33**, D201–D205.

Osterman,A., Overbeek,R. *et al*. (2003) Missing Genes in Metabolic Pathways: A Comparative Genomic Approach. *Chemical Biology*, **7**, 238–251.

Pruitt,K.D., Tatusova,T. and Maglott,D.R. (2005) NCBI Reference Sequence (RefSeq): A Curated Non-redundant Sequence Database of Genomes, Transcripts, and Proteins. *Nucleic Acid Research*, **33**, D501–D504.

Riesenfeld,C.S., Schloss,P.D. and Handelsman,J. (2004) Metagenomics: Genomic Analysis of Microbial Communities. *Annual Review of Genetics*, **38**, 525–552.

Smith,C.R. and Baco,A.R. (2003) Ecology of Whale Falls at the Deep-Sea Floor. *Oceanography and Marine Biology: an Annual Review*, **41**, 311–354.

SoftBerry, FGENESB Suite of Bacterial Operon and Gene Finding Programs, http://www.softberry.com/berry.phtml

Tatusov,R.L., Koonin,E.V. and Lipman,D.J. (1997) A Genomic Perspective on Protein Families. *Science*, **278**, 631–637.

Tringe,S., von Mering,C., Kobayashi,A., Salamov,A., Chen,K. *et al*. (2005) Comparative Metagenomics of Microbial Communities. *Science*, **308**, 554–557.

Tyson,G.W., Chapman,J., Hugenholtz,P. *et al*. (2004) Community Structure and Metabolism through Reconstruction of Microbial Genomes from the Environment. *Nature*, **428**, 37–43.

Venter,J.C., Remington,K., Heidelberg *et al*. (2004) Environmental Genome Shotgun Sequencing of the Sargasso Sea. *Science*, **304**, 66–74.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e368–e374
doi:10.1093/bioinformatics/btl253

An equilibrium partitioning model connecting gene expression and *cis*-motif content

Joe Mellor[1,*] and Charles DeLisi[1]

[1]Program in Bioinformatics, Boston University, Boston, Massachusetts, USA 02215

ABSTRACT

Thermodynamic favorability of transcription factor (TF) binding to DNA is a significant factor in the control of gene expression. Theoretical and *in vitro* measures link the relative equilibrium energy of a particular DNA binding protein to the sequence variation among binding sites in a genome. Extending this principle, we investigate whether biological variation in expression levels of active proteins leads to regulation of different sets of genes, based on inferred affinities of sites upstream of those genes. The TF-concentration-dependent variation in the repertoire of genes regulated by a particular TF is expected to follow patterns of chemical partitioning over DNA sites having differing affinity, and we develop a new modeling approach to test this hypothesis. Based on computational TF binding site discovery and genome-wide expression data available in *Saccharomyces cerevisiae,* we explore motif content for sets of genes and conditions having varying concentrations of different transcription factors which turn those genes on or off. We find cases of significant correlation between the level of intragenomic motif sequence variation and modeled TF protein levels that actuates regulation of corresponding sets of genes, and discuss the observed TF motif variants for several yeast transcription factors, as well as the potential biological functions of genes that are regulated by differential response to these high and low concentrations of particular TFs. These findings suggest that motif sequences of transcription factor binding sites may often be linked with the expression state of corresponding DNA-binding proteins.

Contact: mellor@bu.edu

1 INTRODUCTION

Response to the internal and external cellular millieu is often facilitated by energetically favored binding between at least one regulatory protein and several specific, high affinity consensus motifs in the intergenic DNA. In addition to protein-DNA contacts, transcription factor (TF) binding affinity can be affected by cooperativity, chromatin structure and changes in binding site accessibility. The magnitude of binding affinity at protein-DNA interfaces has been shown to correlate with features including level of sequence variation (1), the presence of multiple motif copies in the intergenic neighborhood (2), and the level of binding found by large-scale chromatin immuno-precipitation (ChIP) experiments (3,4). Transcription factor binding often initiates mechanisms of regulation in RNA production, and while some genes having more primitive regulation mechanisms might escape this paradigm, an understanding of large systems will come in an unraveling of various parts of this regulatory machine.

*To whom correspondence should be addressed.

The biological question we address here is summarized as follows: To what extent are differences in the regulation of various genes mediated by functional differences in a TF's affinity to different upstream intergenic sequences, or the level of the TF needed to bind sets of these sequences? Various forms of evidence suggest that the affinity between proteins and DNA often governs the specificity of regulation (3-11), and we hypothesize that affinity for specific sequences ought to be related to the TF concentration which is needed to turn genes next to those sequences on or off.

Without practical means to address the mechanistic questions directly—that is, to measure *in vivo* affinities for such proteins to all possible motifs in the genome—we instead query for site affinity by proxy. Using a historically observed relationship between motif sequence variation and expected binding affinity, we seek to show how atomistic properties of simple regulatory schemes (*e.g.* TF binding) can be effectively estimated from aggregate measurements of gene expression among system components. We then examine whether the content of *cis*-regulatory elements explains significant differences in responsiveness of downstream genes to various levels of TF. To address these questions we introduce a modeling approach and its application to the *cis*-regulatory sites of the yeast *Saccharomyces cerevisiae*, based on gene expression and upstream sequence information.

Following previous convention (12), a transcription factor's affinity for sequences on DNA can be represented with a position weight matrix (PWM), which indicates preferences for protein binding to any of $b = 4$ possible nucleotides at k independent positions in a set of DNA k-mers. The information content I of the PWM (Eq. 1.1) is a measure of the overall degeneracy (or entropy) of the sequences to which a protein binds.

$$\sum_k \sum_b f(k,b) \frac{\log\left(f(k,b)\right)}{\log\left(f(b)\right)} = I(k,b)_{PWM} \qquad (1.1)$$

Berg and von Hippel showed that, given assumptions of independent contributions of each base at each position in a motif, the PWM equates via statistical thermodynamics to the expected relative free energy ($\Delta\Delta G$) of the binding event at the motif (13), and therefore also to the relative equlibrium binding affinity compared to all possible binding sites in the genome, as shown in Eq. 1.2.

$$I(k,b)_{PWM} \propto \ln(K_{bind}^{eff}) = -\frac{\Delta\Delta G}{RT} \qquad (1.2)$$

The correlation between motif degeneracy and protein-DNA binding free energy leads us to consider whether all motif sites that seem 'allowable' for binding by a transcription factor (that is, with a known PWM) are in fact actually bound by it under

in vivo equilibrium conditions in the cell. Cellular conditions can potentially differ, for example, by having active transcription factor present at different equilibrium concentrations, by having sites that are more or less accessible on the chromosome, or perhaps by post-translational changes to transcription factor activity. The mechanism of regulation is important, as well; for example, some instances of regulation by a protein may require cooperativity with another protein, but other instances may not. To a certain approximation the effective binding constant, K_{bind}^{eff}, of a set of regulatory sites that are *cis* to some collection of genes will, for some fixed concentration of transcription factor protein, determine the fraction of those sites which are bound and likely to involve gene regulation. Similarly, given a constant value of K_{bind}^{eff} across many genes, the ratio of bound to unbound genes might determine the concentration of transcription factor needed to activate those genes. Thus the *in vivo* regulatory program could by this representation be a fairly complex function of the affinity and availability of different *cis* sites, and fluctuations of concentration and interactions between regulating TFs.

We begin with the simplest situation, that of a single protein binding alone to DNA, where the binding site affinity has an inverse relationship with transcription factor concentration at across a range of conditions (TF concentrations) in which regulated genes are bound. This is shown in Eq. 1.3 (assuming uniform concentration of DNA).

$$K_{bind}^{eff} = \frac{[TF]_{bound}}{[TF]_{free}} \tag{1.3}$$

If the amount of bound TF is assumed to be constant across some set of sites near regulated genes, the TF concentration and affinity for those sites should be inversely related. That is, sites with high affinity will be selected on the basis of their thermodynamic favorability at low levels of active TF, but this partitioning between sites will relax at higher concentrations of TF, where sites of lower affinity will bind as well. With a fixed level of bound TF ($[TF]_{bind}$) needed to turn genes on or off, it is possible to consider subpopulations of regulated genes (and their *cis*-sites) which could vary by relative affinity, and therefore potentially respond at relatively higher or lower concentrations of active TF.

Binding (attaining a level of $[TF]_{bound}$) in the simple cases is mediated by the relationship between $[TF]_{free}$ and K_{bind}^{eff} seen in Eq. 1.3. In this study we develop a statistical model to infer whether simple cases such as this exist in the data. We devise a model which addresses whether genes divided into categories modulated by high or low concentrations of a given TF have *cis* sites that can be similarly classified as having 'strong' (high K_{bind}^{eff}) or 'weak' (low K_{bind}^{eff}) affinity. For any fixed set of genes, we assume that the information entropy $I(K, b)$ of *cis* sites near those genes can be used as a proxy for the measurement of K_{bind}^{eff} across those sites, and by Eq. 1.1, we then ask whether sites near those genes turned on or off by high levels of a TF contain different amounts of information than sites near the genes turned on by low levels.

The simple model we propose notwithstanding, it is useful to consider possible alternatives that would deviate from the relation predicted by Eq. 1.3. One such case might be when TF binding to an otherwise low-affinity site is preferred because of an added affinity cause by cooperative binding to another, different regulator. Combinatorial and cooperative mechanisms of transcriptional regulation are abundant in eukaryotes, and in many of these situations, a transcription factor's affinity for a site is potentially mediated by separate, often cooperative, binding events between two or more regulating proteins. Little is known, generally, about the interplay between the affinity of proteins to other proteins or DNA in such cases. Two potentially opposite modes of affinity, one cooperative and the other not, are summarized in Figure 1.

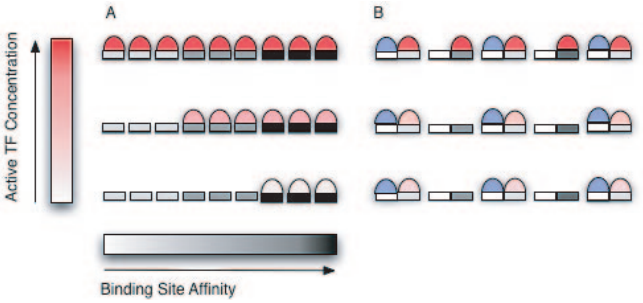


Fig. 1. Binding Site Occupancy Models as a Function of Site Affinity and TF Concentration. (A) In the simplest case, concentration of the active TF (red) controls various gene sets depending on an average affinity of the sites (shades of gray) near of those genes. Under equilibrium conditions, and with many sites, the information content of sites should correspond to their average affinity. (B) Cases of cooperativity between factors possibly deviate from this behavior, when binding is mediated by an additional protein (blue) which binds nearby, changing the effective affinity of the sites. The relationship between motif content, TF concentration and site occupancy could be altered to favor low affinity sites at low TF concentration.

DNA sites with affinity for cooperative proteins are not necessarily bound solely based on their intrinsic affinity, or the availability of the active TF, but also due to favorable TF binding to other proteins, and of these proteins to other, neighboring sites. In cases where the thermodynamic selection for a ''weak'' site by a protein is preferred because that site neighbors a site of another cooperating protein to which the first protein binds, ''Strong'' sites, on the other hand, which have higher affinity, can bind TFs in the absence of the cooperating protein. This type of 'neighboring site effect' has been recently shown to play a role in governing the content of *cis* elements for several TFs in yeast(5).

While combinatorial effects may be common among eukaryotic mechanisms of regulation, TFs and *cis*-sites near genes obeying the simpler relationship of Eq 2 (*i.e.*, Figure 1a) are still quite interesting. First, they represent cases where the dynamics of regulatory behavior are approximately reducible to TF concentrations and estimated binding strengths alone. Second, these are cases where regulation mechanisms are possibly more accessible to efforts in engineering of synthetic systems; their relatively simple behavior makes them ideal for manipulation in novel systems.

2 METHODS

A yeast TF is designated as either an activator or repressor based on primary evidence collected in the *Saccharomyces* Genome Database (14). As is the case in many previously described expression-based models (15-17), we'll assume transcription of genes by this TF can be used as a predictor of the TFs protein-level activity. Though the strength of this predictor may vary by TF, we'll assume it is a uniform predictor regardless of what genes a particular TF regulates. We'll also posit that the protein-level activity of a TF can be ascertained in many cases from its own gene expression data. Simplifications

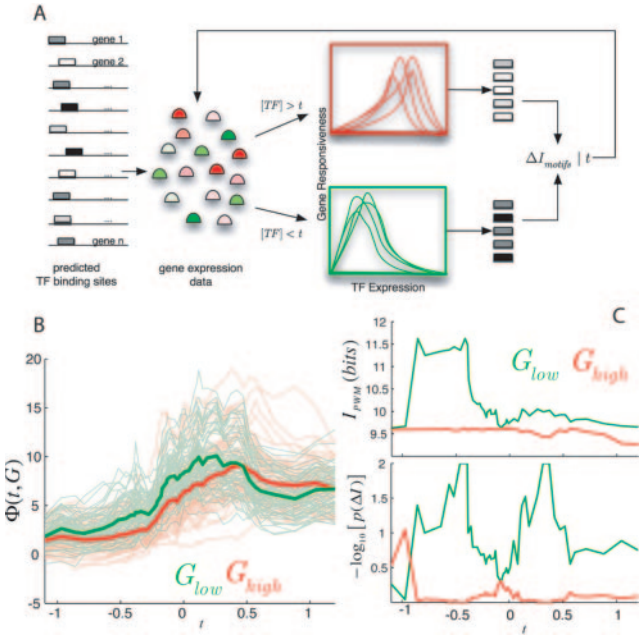


Fig. 2. Regulation Modeling Procedure and Analysis of Expression-Based Motif Sets. **(A)** Sets of genes with motifs for a given transcription factor are analyzed to produce an estimate of the difference in motif information content as a function of the TF expression threshold parameter *t*. The method identifies cases where the divergence in motif content corresponds to a divergence in regulatory response to the TF. **(B)** An example of the regulation scoring function profile for genes with at least one motif for the amino acid biosynthesis regulator *GCN4*, for *t=0.25*. Genes in ''high'' group (red) are maximally responsive to levels of *GCN4* greater than *t*; genes in the ''low'' group (green) to levels less than *t*. **(C)** The upper plot shows information content (in bits) of motifs in high and low gene groups as a function of *t*. Lower half shows the probability of observing a difference in information as great as that between low and high groups at values of *t*.

such as this will fail to account for some quite relevant biological situations, for example where a TF is constitutively expressed or active, where the level of TF or target activity is controlled exclusively by post-translational modifications, or where regulation in general is not mediated by mRNA production. Using mRNA measurements alone potentially underestimates, but not likely contradicts, the importance of these more complex mechanisms in our model. And while the model is much simpler than a fuller physical model of the processes involved, it usefully isolates variables which can be easily measured by experiments such as gene expression arrays and protein-DNA ChIP assays.

2.1 Computation of gene regulatory models

The outline of our procedure is summarized in Figure 2. We begin with a collection of hypotheses, or putatively independent regulatory models, that follow from any pair of genes *A* and *G*, where a high-scoring sequence element for a motif representing transcription factor *a* (encoded by *A*) is found by motif scanning in the 1000 bp region upstream of the start codon of *G*. The set $\mathbf{G}_A = \{G_1, G_2, \ldots, G_n\}$ then represents the collection of such genes for a TF *A*; we then ask whether expression exhibited by the transcription factor expression is significantly associated with a net activation or repression of the target gene. Transcription factor binding sites (TFBS) were predicted from PWMs derived for a number of yeast transcription factors based on intergenic binding (ChIP) in the recent paper by Harbison, *et al* (3). We used MotifScanner (18) with these matrices to scan 1000 bp regions in the 5′ upstream area of each open reading frame in the *Saccharomyces cerevisiae* genome. Each gene hit, and the corresponding motif sequences in the input to the expression analysis.

From a set of expression measurements (two-color microarrays), we denote expression values for a TF *A* and gene *G* as the joint distribution $\theta_{A,G}$. We used the Rosetta deletion compendium of ∼300 expression conditions (19), normalizing the expression of each gene in this set to have zero mean and unit variance. We use a suitable function $\Phi(t_A, G)$ (Eq 2.1) to represent the magnitude of activation or repression of *G* with respect to different expression values of *A*. We denote this function as the log-likelihood ratio that some expression change in *G* occurs when the expression of *A* exceeds a threshold value $t_A(G)$.

$$\Phi(t_A, G) = -\delta \log_{10}\left(\frac{p[(\theta_G \mid \theta_{A>t_A}) < (\theta_G \mid \theta_{A<t_A})]}{p[(\theta_G \mid \theta_{A>t_A}) > (\theta_G \mid \theta_{A<t_A})]}\right) \tag{2.1}$$

Values obtained by Eq 2.1 are calculated using bootstrap samplings from the distribution of gene *G*, from conditions where the normalized expression level of TF *A* is either above or below a threshold value t_A. Probability terms representing the differences in the conditional distributions of *G* are calculated using the Kolmogorov-Smirnov test. A sign parameter $\delta = (-1, +1)$ indicates whether the TF is a known to be an activator (+1) or repressor (−1), based on literature annotation—high values of $\Phi(t_A, G)$ indicate significant shifts in *G* expression consistent with the TF's known function.

We estimate the optimal threshold $t_A(G)$ that maximizes the value of the statistic Φ for gene *G* by calculating $\Phi(t_A, G)$ for many values of $t_A(G)$ based on 100 bootstrapped samplings from the original expression data. Results of this sampling procedure are then aggregated to produce a mean value of the scoring statistic $\bar{\Phi}(t_A, G)$, corresponding to an average value of the threshold parameter $\bar{t}_A(G)$. The score $\bar{\Phi}(t_A, G)$ then represents an optimum in the average 'responsiveness' of gene *G* to the expression level of gene *A* at expression level $\bar{t}_A$, but doesn't imply that *G* is only regulated at this level. Because we average over many random selected models created from the data, the estimate of $\bar{\Phi}(t_A, G)$ and $\bar{t}_A$ summarizes the most persistent effects over many perturbations and conditions in the original data.

2.2 Computation of TFBS partitioning

Each TF and gene combination produces a model fit by the above procedure, with varying values of $\bar{\Phi}(t_A, G)$ and $\bar{t}_A$. Because the motif scanning procedure exhibits known low specificity, and because expression values between unrelated gene can be weakly correlated at random, it is appropriate to filter the results of the expression-based modeling to select combinations for which both motif and regulation data are both present. Numerous iterations of the procedure found $\bar{\Phi}(t_A, G) = 3.0$ to be an effective value for removing spurious associations at minimal cost to further analysis. This filtered set of modeled genes is $\mathbf{G}_A^{filtered} = \{G_1, G_2, \ldots, G_n\}$. In general, any over-fitting which is likely encountered by having many multiple independently parameterized models is avoided by using models averaged over many sets of conditions, and as we describe next, many sets of genes.

The next step of our procedure seeks to partition or permute the filtered set of modeled genes, $\mathbf{G}_A^{filtered}$, into sets having significant differences in *cis* motif content. These sets will be explicitly dependent on modeled values of $\bar{t}_A$, which are obtained in the previous stage. By selecting genes having average modeled values of $\bar{t}_A$ greater than (*high*) or less than (*low*) some threshold value t_A^{crit}, we produce two new sets $\mathbf{G}_A^{high}$ and $\mathbf{G}_A^{low}$, whose aggregate maximum $\bar{\Phi}(t_A, G)$ occurs at relatively high and low levels of TF relative to t_A^{crit}. Each value of t_A^{crit} yields a new pair of sets of aligned motifs *cis* to the genes in $\mathbf{G}_A^{high}$ and $\mathbf{G}_A^{low}$, from which the information weight matrix (*I*) can be calculated by using the PWM of each set of motifs:

$$I_{PWM} = \sum_l \sum_b f(b,l) \log_2 (f,l) \tag{2.2}$$

This process is then repeated for successive values of t_A^{crit}. In addition, providing a more robust estimate the mean and variance of the information content from substituent motifs in the gene set, we additionally bootstrap the

selection of the substituent genes (x100) from both high and low sets at each chosen value of $\bar{t}$. The value of t_A^{crit} producing the maximum bootstrap significance (bootstrap $p \leq 0.01$) in PWM information difference between set $\mathbf{G}_A^{high}$ and $\mathbf{G}_A^{low}$ defines two optimal average PWMs (PWM_{high} and PWM_{low}) for the two sets. Finally, a more precise estimate of the average information difference $\Delta I_A = I(\mathbf{G}_A^{low}) - (\mathbf{G}_A^{high})$between motifs in the sets, and the significance of this final difference is estimated from a last much larger bootstrapped sampling (x1000) of gene sets having equal size as final sets $\mathbf{G}_A^{high}$ and $\mathbf{G}_A^{low}$ made from the original set $\mathbf{G}_A^{filtered}$. This probability estimate final gives the level of surprisal that the information content of PWM_{high} and PWM_{low} would diverge as much as is observed over many random samplings from our original large set of motifs.

The output for each TF, therefore, the optimal difference in PWM information entropy, ΔI_A as a function of the threshold of expression for the TF,$\bar{t}_A$. The process implicitly yield two sets of genes, $\mathbf{G}_A^{high}$ and $\mathbf{G}_A^{low}$, whose regulation appears to respond in aggregate to correspondingly high or low expression levels of TF *A*. As noted earlier, an increase in this measure of information among sets of protein-bound sequences corresponds to an increase in the statistical thermodynamic estimate of a relative binding affinity between the protein and DNA, K_{bind}^{eff}. The procedure therefore gives us way to examine the relationship between the modeled concentration of TF needed to regulate its genes, and the affinity of sites upstream of those genes.

3 RESULTS

We applied the outlined procedure to several yeast transcription factors for which PWM data was sufficiently available. We excluded transcription factors for which we could find no sites, or TFs for which predicted sites were identical across all genes; obtaining informative subsets is impossible in these cases. The results for seven TFs are shown in Table 1.

3.1 Site content and transcription factor level at transcriptionally active genes

We were able to recover a number of instances where the information content of TFBS motifs could be partitioned into significant subsets depending on the modeled level of transcription factor which corresponded to activation of different genes in these sets. A surprising result of the study was that most transcription factors (10 out of 12 cases) showed some type of significant change in information content ($p \leq 0.05$) as a function of the modeled value of t_A. The probabilities reported in Table 1 are not Bonferroni correct, however, and therefore possibly of marginal significance when judge in total, but the individual probabilities of a several cases are at 0.01 or better, and have supporting evidence as we discuss further. Based on this evidence it is likely that TF expression corresponds in some cases to changes in the information of *cis* sites on genes, but it apparent that this change can be positive or negative.

Half of the significant cases we found (5 of 10), the change in information was positive – Gcn4, Leu3, Hap1, Msn4 and Skn7 made up this set. The remaining five (Rpn4, Mcm1, Swi4, Abf1 and Ume6) had significant negative change in PWM information content between 'high' and 'low' TF-regulated gene sets. Based on the predictions of our basic model, we expect cases where the change in information is positive between high versus low TF levels to reflect those situations where concentration and binding site affinity are dominant in governing regulation. The other cases show an opposite tendency, suggesting certain TFs regulate genes with lower site affinity even at lower TF levels, perhaps due to cooperativity.

The two transcription factors shown in Table 1 (Mcm1 and Rpn4) which exhibit strong negative change in gene TFBS information content (ΔI) at low TF levels are both known participants with other factors during DNA binding and regulation. Rpn4p participates in a proteosomal auto-regulation pathway (20), while Mcm1p alternately binds to alpha1, alpha2 and Ste12p during different stages of mating and mate-type gene expression (21). The preference of Rpn4 for high-affinity sites at higher concentrations could be linked to the observed negative feedback mechanism whereby the targets of Rpn4 encode proteins that ultimately degrade the TF itself. The transcription factors which exhibit positive change in motif information content, Gcn4, Leu3, Hap1, Msn4 and Skn7 are generally non-complexed protein regulators of gene expression, although some (*e.g.* Gcn4p and Hap1p) are known to dimerize before binding DNA.

The results of the modeling procedure are suggest that potential binding mechanisms can be seen in the increased preference for certain nucleotides in averaged PWMs corresponding to low and high information sets. For example, in Skn7 sites, the additional information gained between site partitions is also entirely due to the conservation of C at position 2. The Rpn4 motif shows an increased preference for a triplet G in the beginning of the motif, and Hap1 sites show increased conservation in the middle positions between two highly conserved ends.

3.2 Biological significance of motif partitioning

Our approach assumes that the affinity of transcription factor sites can be functionally segregated based on regulatory pattens among their downstream *cis* genes. If this is true, then we should expect cases where a biological or functional interpretation can be associated with this type of partitioning. As shown in Table 1 we investigated high and low sets of modeled genes for each TF to see if they were particularly enriched for binding in large-scale ChIP or for functional information in the Gene Ontology. These tests potentially provide indirect evidence that the genes sets of different sets exhibit different specificities in binding assays, or are involved in different types of cellular processes. We report for each high and low set the most-enriched TF in ChIP binding assays and the most common functional category in the Gene Ontology. In some cases (*Gcn*4, *Mcm*1, *Msn*4) ChIP results returned the identical TF for least one of the sets. For many other cases, however, the results returned other high-scoring TFs, suggesting that the binding of these different TFs may have cross-specificity or target gene overlap. Whether this represents binding of related TFs to the same or similar upstream sequences in ChIP experiments, or the binding of TFs to each other, remains to be explored.

Different gene sets obtained by our partitioning method are often enriched for separate GO categories despite sharing the same TF, suggesting some degree of functional heterogeneity exists among genes responsive to different TF levels. For example Skn7 targets were most represented in metabolism (high *SKN*7 expression) and biogenesis (low *SKN*7 expression). Often, at least one of the gene sets for each TF represented functional categories that correspond to processes regulated by the TF. For example Leu3 target genes are involved in amino acid biosynthesis (22), Rpn4 genes in proteolysis (20), Gcn4 genes in amino acid biosynthesis (23), Hap1 in catalytic activity (24), *etc*., are consistent with literature evidence of function for these regulators.

3.3 Biological validation of motif variants

We further analyzed examples found as part of this study to see if supporting evidence exists for the binding and function of motif

Table 1. Summary of modeled PWM content relating to transcriptional activities

TF	$\langle PWM_{high} \rangle$	$\langle I_{high} \rangle$ (bits)	top ChIP^2 % (prob) top GO (prob)	$\langle PWM_{low} \rangle$	$\langle I_{low} \rangle$ (bits)	top ChIP^2 % (prob) top GO (prob)	$\Delta\langle I_{eff} \rangle$ (bits)	$\Delta\langle t \rangle$	prob (ΔI_{eff}) (random)
Gcn4		8.29 ± 0.18	***GCN4* 52.8% (0.008)** amino acid biosynthesis (5.3e-05)		9.27 ± 0.21	***GCN4* 19.5% (6.5e-12)** amino acid biosynthesis (1.6e-21)	**0.94 ± 0.28**	0.77 ± 0.45	0.005
Rpn4		12.88 ± 0.34	***ARG81* 11.1% (0.08)** proteolysis (0.02)		11.75 ± 0.19	***NRG1* 10.9% (0.02)** transcription factor complex (0.0008)	**-1.13 ± 0.39**	1.51 ± 0.49	0.007
Leu3		8.43 ± 0.14	***RIM101* 2% (0.005)** amino acid metabolism (0.01)		9.31 ± 0.34	***RPN4* 8.8% (0.009)** transport (0.0007)	**0.96 ± 0.36**	0.69 ± 0.26	0.015
Hap1		13.50 ± 0.17	***GCN4* 9.5% (0.002)** catalytic activity (7.0e-05)		14.71 ± 0.51	***TEC1* 5.0% (0.02)** catalytic activity (0.05)	**1.15 ± 0.55**	1.38 ± 0.75	0.011
Mcm1		11.66 ± 1.65	***NDD1* 27.3% (0.004)** site of polarized growth (0.02)		9.08 ± 0.20	***MCM1* 10.9% (5.0e-5)** helicase activity (5e-06)	**-2.76 ± 1.66**	2.05 ± 0.83	0.025
Msn4		10.27 ± 0.47	***CUP9* 5.0% (0.05)** carbohydrate metabolism (0.002)		11.11 ± 0.20	***MSN4* 14.1% (0.01)** generation of precursor metabolites and energy (5e-10)	**0.77 ± 0.50**	0.93 ± 0.38	0.041
Skn7		12.42 ± 0.41	***XBP1* 5.8% (0.004)** metabolism (0.0001)		13.46 ± 0.46	***IFH1* 6.7% (0.06)** cytoplasm organization and biogenesis (6e-06)	**1.08 ± 0.48**	1.51 ± 0.77	0.011

Sequence logos depict the model-averaged position-weight matrices (PWMs) for sets of genes whose regulation occurs at corresponding high and low expression of the transcription factor (TF). Mean information content of motifs in the ''high'' and ''low'' sets are given, along with the probability of measuring the observed information difference in bootstrap randomized subsets of gene sets of equal size. Measured enrichment corresponding to binding in large-scale ChIP assays (3,4), as well as the most enriched category (30) in the Gene Ontology (31), is reported for both gene sets. ΔI and Δt are the mean PWM information content and average TF expression level among modeled genes in the low minus the high sets.

variants in experimental literature. We were able to find evidence to support three of the observed motifs for TFs partitioned on the basis of regulatory activity.

Leu3
An experimental *in vitro* selection assay of dissociation constants (K_D) of 43 variants of the binding site of Leu3 was recently performed by Liu and Clarke (25), who found that several variants had higher affinity than the accepted consensus motif for this TF ('CCGGTACCGG'). The classical description of the Leu3p binding site is of two everted 'CCG' repeats separated by four nucleotides. The PWM for motifs found by scanning the yeast genome indicate that the majority of positional information is indeed found in the repeats, which are bound by two domains of the leucine zipper Leu3, regardless of the gene group targets. Our analysis predicted that a higher affinity version of this motif (Table 1) has increased positional information at the two bases at positions four and five immediately after the first repeat. The binding experiments by Liu and Clarke confirms the affinity conferred by a T at position four has over three-fold higher affinity than the consensus motif. The other T, at position five, has marginally stronger affinity than the consensus G reported for that position in the Leu3 site.

Hap1
Hap1 is a member of the Zn_2Cys_6 family of binuclear cluster TFs which bind as homodimers, typically to CGG repeats. Our analysis of Hap1 binding sites predicted that sites bound by higher levels of expression of Hap1 have enrichment for information in the spacer region at positions 3 to 8. This observation is corroborated by a recent experimental analysis of the composition of the Hap1p binding site performed by Wang, *et al* (26), who found that deletion of the spacer region between repeats lowered the effective affinity of the TF for sequences *in vitro*. They also report a binding preference for a dinucleotide TA in this spacer region.

Gcn4
The sequence specificity of the bZip family member Gcn4 has been previously reported to bind to a half site in DNA (27) with a consensus seven base-pair sequence 'TGA(C/G)TCA'(28). The preference for T and G at the first and second positions of this core, and for A at the last position, are more variable among sites in the yeast genome than the four base pairs at positions three through six. This observation agrees with the prediction made by our analysis among gene sets regulated by high and low levels of Gcn4 expression; specifically, we find there is a preference for the canonical binding motif among gene regulated at lower levels of the transcription factor, and this is weakened among gene regulated at higher levels of Gcn4 expression. An overall difference in PWM information content of 0.94 ± 0.28 (bits), as seen in Table 1, is due largely to a preference for TG at the first two positions in the core motif, as well as A at the last position.

4 DISCUSSION

The equilibrium partitioning of transcription factors among sites of different binding affinity across the genomes is a simple but potentially important mechanism that plausibly controls whole sets of genes across different conditions. Using a novel, thermodynamically motivated approach, we've presented preliminary evidence based on predicted TF binding sites and expression data in *Saccharomyces cerevisiae* that this general effect might play some role between certain transcription factors and their targeted genes. In these cases, a significant inverse relationship was noted between the aggregate information content of a set of motifs, and the expression level of the TF that putatively binds these motifs. Gene regulation in these cases is plausibly tuned to respond to various conditions by the interplay between available transcription factors and the affinity of sites to which they bind.

Our results also suggest that site affinity plays a more complicated role in the specificity of TFs acting in tandem, however, where the affinity of sites is possibly dependent on the activity of other proteins that can bind at or near the same region of DNA. The cases where we find strong evidence that regulation follows a simpler model are potentially more attractive targets for forward engineering in synthetic systems.

The analysis we've shown doesn't prove the actual physical affinities of partitioned sites, and for this further experiments will clearly be necessary. A variety of assumptions must be made in linking the information in the sites with the biophysical interpretation of binding preferences, but the observed correlations between modeled target gene activity and motif content lend support to the model we've used, and suggest that functional knowledge of biological systems can be gained by this simplification. There are also clear limitations where binding site partitioning, whether statistical or thermodynamic, is effectively impossible, for example if a particular site is completely invariant across all genes or has uniform affinity.

We note that, in general, more complicated cooperative effects aren't incompatible with the model of binding and regulation we describe here, and in fact these cases might adequately be described as modulations to the single protein equilibrium. These modulations can be obtained by changing the effective affinity of proteins for sites secondary binding events, having protein co-localization, co-orientation, *etc*. Signatures of cooperativity might therefore be detectable as in the content of combined *cis* regulatory signals, as well, or the expression and activity of combinations multiple TFs, leading to an enriched understanding of regulatory logic. The basic model we've used here can be extended to consider combinations and sets of transcription factors, where the affinity of combinations of TFs is influenced by motif content, as well as relative orientation. In such cases, factors other than the information content of motifs may play a much more important role. In work along such lines, Beer and Tavazoie (29) showed that spatial patterns in motif organization are sufficient to predict the regulatory response of many genes. The extraction of mechanistic rules in systems of combinatorial regulation is a remaining challenge for this and many other modeling approaches.

To summarize, despite a of lack direct methods to (a) verify the *in vivo* affinity of any particular *cis* sites in the genome, (b) to understand mechanisms of affinity at arbitrary protein-DNA interfaces, (c) know the effective protein concentrations of different active TFs, or (d) measure the availability of TF sites on DNA, we can still approach some of these questions with modeling methods based on available data. In this study we examined the relationships between transcription factor affinity and regulatory efficacy by using model based on an assumed physio-chemical partitioning that occurs during binding and regulation. In testing this model, and whether genes are aggregately activated or repressed in response to

high and low levels of a transcription factor's expression, we found interesting signatures of the dynamical processes involved in gene regulation. These patterns are in several cases sufficient to identify significant and functional differences between *cis*-elements to which a transcription factor binds.

ACKNOWLEDGEMENTS

The authors thank Melissa Landon, Evan Snitkin and Yaoyu Wang for helpful discussions.

REFERENCES

Moses,A.M., Chiang,D.Y., Kellis,M., Lander,E.S. and Eisen,M.B. (2003) Position specific variation in the rate of evolution in transcription factor binding sites. *BMC Evol Biol*, **3**, 19.

Frith,M.C., Li,M.C. and Weng,Z. (2003) Cluster-Buster: Finding dense clusters of motifs in DNA sequences. *Nucleic Acids Res*, **31**, 3666–3668.

Harbison,C.T., Gordon,D.B., Lee,T.I., Rinaldi,N.J., Macisaac,K.D., Danford,T.W., Hannett,N.M., Tagne,J.B., Reynolds,D.B., Yoo,J. *et al.* (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**, 99–104.

Lee,T.I., Rinaldi,N.J., Robert,F., Odom,D.T., Bar-Joseph,Z., Gerber,G.K., Hannett,N.M., Harbison,C.T., Thompson,C.M., Simon,I. *et al.* (2002) Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, **298**, 799–804.

Bilu,Y. and Barkai,N. (2005) The design of transcription-factor binding sites is affected by combinatorial regulation. *Genome Biol*, **6**, R103.

Nachman,I., Regev,A. and Friedman,N. (2004) Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, **20 Suppl 1**, I248–I256.

Granek,J.A. and Clarke,N.D. (2005) Explicit equilibrium modeling of transcription-factor binding and gene regulation. *Genome Biol*, **6**, R87.

Djordjevic,M., Sengupta,A.M. and Shraiman,B.I. (2003) A biophysical approach to transcription factor binding site discovery. *Genome Res*, **13**, 2381–2390.

Benos,P.V., Bulyk,M.L. and Stormo,G.D. (2002) Additivity in protein-DNA interactions: how good an approximation is it? *Nucleic Acids Res*, **30**, 4442–4451.

Bulyk,M.L., Johnson,P.L. and Church,G.M. (2002) Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Res*, **30**, 1255–1261.

Mukherjee,S., Berger,M.F., Jona,G., Wang,X.S., Muzzey,D., Snyder,M., Young,R.A. and Bulyk,M.L. (2004) Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. *Nat Genet*, **36**, 1331–1339.

Schneider,T.D., Stormo,G.D., Gold,L. and Ehrenfeucht,A. (1986) Information content of binding sites on nucleotide sequences. *J Mol Biol*, **188**, 415–431.

Berg,O.G. and von Hippel,P.H. (1987) Selection of DNA binding sites by regulatory proteins Statistical-mechanical theory and application to operators and promoters. *J Mol Biol*, **193**, 723–750.

Cherry,J.M., Adler,C., Ball,C., Chervitz,S.A., Dwight,S.S., Hester,E.T., Jia,Y., Juvik,G., Roe,T., Schroeder,M. *et al.* (1998) SGD: Saccharomyces Genome Database. *Nucleic Acids Res*, **26**, 73–79.

Friedman,N., Linial,M., Nachman,I. and Pe'er,D. (2000) Using Bayesian networks to analyze expression data. *J Comput Biol*, **7**, 601–620.

Pe'er,D., Regev,A., Elidan,G. and Friedman,N. (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17 Suppl 1**, S215–224.

Pilpel,Y., Sudarsanam,P. and Church,G.M. (2001) Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat Genet*, **29**, 153–159.

Aerts,S., Thijs,G., Coessens,B., Staes,M., Moreau,Y. and De Moor,B. (2003) Toucan: deciphering the cis-regulatory logic of coregulated genes. *Nucleic Acids Res*, **31**, 1753–1764.

Hughes,T.R., Marton,M.J., Jones,A.R., Roberts,C.J., Stoughton,R., Armour,C.D., Bennett,H.A., Coffey,E., Dai,H., He,Y.D. *et al.* (2000) Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.

Xie,Y. and Varshavsky,A. (2001) RPN4 is a ligand, substrate, and transcriptional regulator of the 26S proteasome: a negative feedback circuit. *Proc Natl Acad Sci U S A*, **98**, 3056–3061.

Zhong,H., McCord,R. and Vershon,A.K. (1999) Identification of target sites of the alpha2-Mcm1 repressor complex in the yeast genome. *Genome Res*, **9**, 1040–1047.

Friden,P. and Schimmel,P. (1988) LEU3 of Saccharomyces cerevisiae activates multiple genes for branched-chain amino acid biosynthesis by binding to a common decanucleotide core sequence. *Mol Cell Biol*, **8**, 2690–2697.

Hinnebusch,A.G. (2005) Translational regulation of GCN4 and the general amino acid control of yeast. *Annu Rev Microbiol*, **59**, 407–450.

Becerra,M., Lombardia-Ferreira,L.J., Hauser,N.C., Hoheisel,J.D., Tizon,B. and Cerdan,M.E. (2002) The yeast transcriptome in aerobic and hypoxic conditions: effects of hap1, rox1, rox3 and srb10 deletions. *Mol Microbiol*, **43**, 545–555.

Liu,X. and Clarke,N.D. (2002) Rationalization of gene regulation by a eukaryotic transcription factor: calculation of regulatory region occupancy from predicted binding affinities. *J Mol Biol*, **323**, 1–8.

Wang,L.L., Denman,I. and Junker,M. (2004) Control of Hap1-DNA site recognition through the interplay of multiple distinct intermolecular interactions. *Biochemistry*, **43**, 13816–13826.

Hollenbeck,J.J. and Oakley,M.G. (2000) GCN4 binds with high affinity to DNA sequences containing a single consensus half-site. *Biochemistry*, **39**, 6380–6389.

Oliphant,A.R., Brandl,C.J. and Struhl,K. (1989) Defining the sequence specificity of DNA-binding proteins by selecting binding sites from random-sequence oligonucleotides: analysis of yeast GCN4 protein. *Mol Cell Biol*, **9**, 2944–2949.

Beer,M.A. and Tavazoie,S. (2004) Predicting gene expression from sequence. *Cell*, **117**, 185–198.

Boyle,E.I., Weng,S., Gollub,J., Jin,H., Botstein,D., Cherry,J.M. and Sherlock,G. (2004) GO::TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics*, **20**, 3710–3715.

Harris,M.A., Clark,J., Ireland,A., Lomax,J., Ashburner,M., Foulger,R., Eilbeck,K., Lewis,S., Marshall,B., Mungall,C. *et al.* (2004) The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res*, **32**, D258–261.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e375–e383
doi:10.1093/bioinformatics/btl202

Identification of metabolic units induced by environmental signals

Jose C. Nacher*, Jean-Marc Schwartz*, Minoru Kanehisa and Tatsuya Akutsu

Bioinformatics Center, Institute for Chemical Research, Kyoto University Uji, Kyoto 611-0011, Japan

ABSTRACT

Motivation: Biological cells continually need to adapt the activity levels of metabolic functions to changes in their living environment. Although genome-wide transcriptional data have been gathered in a large variety of environmental conditions, the connections between the expression response to external changes and the induction or repression of specific metabolic functions have not been investigated at the genome scale.

Results: We present here a correlation-based analysis for identifying the expression response of genes involved in metabolism to specific external signals, and apply it to analyze the transcriptional response of *Saccharomyces cerevisiae* to different stress conditions. We show that this approach leads to new insights about the specificity of the genomic response to given environmental changes, and allows us to identify genes that are particularly sensitive to a unique condition. We then integrate these signal-induced expression data with structural data of the yeast metabolic network and analyze the topological properties of the induced or repressed subnetworks. They reveal significant discrepancies from random networks, and in particular exhibit a high connectivity, allowing them to be mapped back to complete metabolic routes.

Contact: nacher@kuicr.kyoto-u.ac.jp, jean@kuicr.kyoto-u.ac.jp

1 INTRODUCTION

Complex interactions between multiple molecular compounds and mechanisms are responsible for cellular functions. Huge amounts of experimental data have allowed a growth in knowledge about biochemical processes and interactions, but the integration of all these data in order to reach a global understanding of the behavior of a biological cell is just starting. Metabolic processes are a key element of cellular behavior, and the analysis of metabolic networks has therefore gained much attention in recent years. Many efforts have concentrated on the structural analysis of metabolic networks (Jeong *et al*., 2000; Wagner and Fell, 2001; Almaas *et al*., 2004) and new methods such as elementary mode and extreme pathway analyses have been developed (Schuster *et al*., 2000; Schilling *et al*., 2000). Genome-scale models of metabolism have been reconstructed for a growing number of organisms (Edwards *et al*., 2000; Förster *et al*., 2003; Ma and Zeng, 2003). However, the metabolic network of a biological organism is a highly dynamically regulated system, and structural analysis alone is not sufficient. In order to understand the dynamical activity of metabolic processes and the mechanisms regulating this activity, structural analysis of metabolic networks needs to be combined with other sources of information, such as gene expression data.

Genome-scale expression analyses are now routinely performed for a wide range of experimental conditions, and many tools are available for the analysis of expression data and the identification of statistically significant increases or decreases in gene expression. Changes in expression levels have sometimes been mapped to precise metabolic pathways (DeRisi *et al*., 1997; Ideker *et al*., 2001; Krömer *et al*., 2004; Zaslaver *et al*., 2004), but the relations between the differential expression of some genes and the activation or repression of metabolic routes have not been investigated at the cellular level. Approaches have been presented for the identification of sets of genes contributing to the same metabolic pathway and whose expression levels are coordinated to a particular phenotype (Barriot *et al*., 2004; Lee *et al*., 2005; Tian *et al*., 2005). However, these approaches do not allow the mapping of these gene sets to particular metabolic processes or the identification of connected metabolic routes. In parallel, several tools have been developed for visualizing expression data on metabolic pathways (Dahlquist *et al*., 2002; Borisjuk *et al*., 2004; Mlecnik *et al*., 2005), highlighting the interest in combining these two sources of data for understanding the organization and dynamical evolution of cellular processes.

The yeast *Saccharomyces cerevisiae* is well suited to such integrative analyses, firstly because it is one of the most thoroughly studied organisms, with the structure of its metabolic network as well as the functions of many genes being well known, and secondly because it has evolved to be able to survive rapid and drastic changes in its environment. Unicellular organisms must be able to rapidly adjust their internal systems to fluctuations in external conditions, and one aspect of this adaptation is the reorganization of genomic expression to each new environment (Gasch and Werner-Washburne, 2002). Recently, it has been shown that in *E. coli* distinct transcriptional subnetworks are responsible for environmental perturbation processing (Balázsi *et al*., 2005), and some approaches have been developed for studying the transcriptional regulatory architecture of metabolic networks (Guelzim *et al*., 2002; Patil *et al*., 2005). However, the coupling between multiple environmental changes and the induction (repression) of specific metabolic routes has not been investigated. In this paper, we first present a correlation-based analysis of gene expression patterns corresponding to various environmental conditions. We show that this approach leads to new insights about the specificity of

*These authors contributed equally to this work.
To whom correspondence should be addressed.

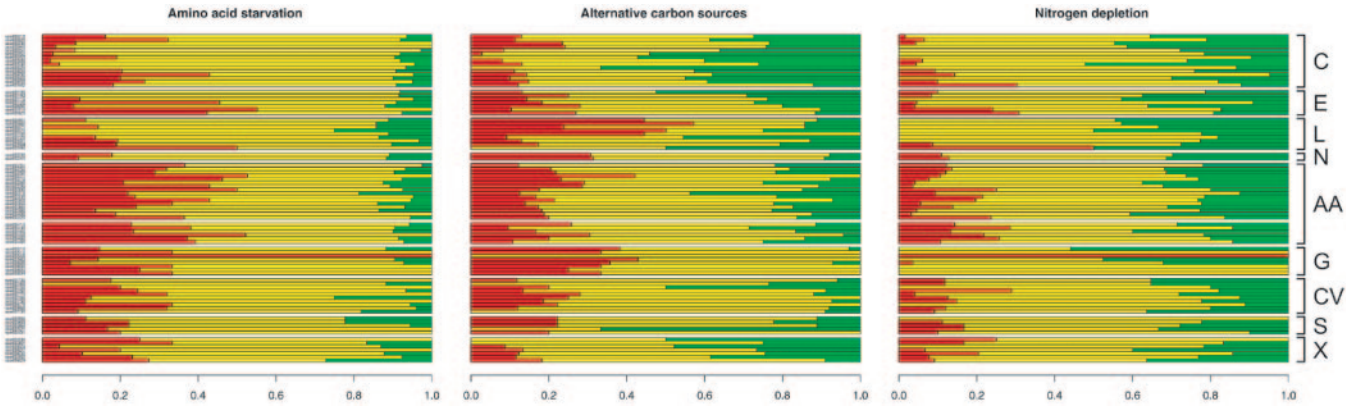


Fig. 1. Distribution of induced (repressed) genes in metabolic pathways. The fraction of significantly induced genes ($z_i^{(s)} > 1$) is displayed in red, that of significantly repressed genes ($z_i^{(s)} < -1$) in green, and that of non-significantly affected genes in yellow. Each line corresponds to one metabolic map in the KEGG database, identified by the numbers on the left-hand side. Pathways are classified into the following categories as in the KEGG database: (C) carbohydrate metabolism, (E) energy metabolism, (L) lipid metabolism, (N) nucleotide metabolism, (AA) amino acid metabolism, (G) glycan biosynthesis and metabolism, (CV) metabolism of cofactors and vitamins, (S) biosynthesis of secondary metabolites, (X) biodegradation of xenobiotics.

the genomic response to given environmental conditions, and allows us to identify genes that are particularly sensitive to a unique condition. We then present an integration of such signal-induced expression data with structural data of the yeast metabolic network. We show that the sets of genes that are significantly induced (respectively repressed) in a given condition build connected subnetworks. Characterization of the topological properties of these subnetworks reveals significant discrepancies from random networks, in particular a higher connectivity, allowing them to be mapped back to complete metabolic routes.

2 RESULTS

2.1 Genomic response is coupled to specific environments

The first part of this study consisted of looking for genes whose expression response is strongly coupled to specific environmental conditions. Such an analysis requires the availability of expression data in many different conditions, as well as a rich amount of data for each condition. We used microarray data obtained from experimental work by Gasch *et al.* (2000) for 6152 genes of *S. cerevisiae* in 13 different environmental shocks. For each condition, we defined a binary signal representing an idealized expression pattern that is fully correlated to the given condition (see Methods). Then, expression patterns of all genes were compared to this idealized pattern to identify genes with the strongest positive and negative correlations. This process led to the computation of individual $z_i^{(s)}$ scores, quantifying the strength of coupling of the expression of each gene i to each condition s.

These genes were mapped to metabolic pathways using the KEGG database (Kanehisa *et al.*, 2006). 819 out of the 6152 genes present in the microarray data were found to be involved in yeast metabolism. As each pathway map in KEGG corresponds to a particular biological functionality, we analyzed the distribution of $z_i^{(s)}$ scores in the 85 yeast-specific metabolic maps from KEGG for each of the 13 environmental conditions.

The distribution of $z_i^{(s)}$ scores was found to be significantly specific to each condition (Figure 1). Temperature shocks, diamide treatment and amino acid starvation produced larger numbers of induced genes in many parts of metabolism, while nitrogen depletion or stationary phase produced more repressed genes (Table 1). In other cases, the dominant response varied depending on the area of metabolism: for example, stationary phase experiments showed a dominance of induction among genes involved in carbohydrate metabolism and a dominance of repression among genes involved in amino acid metabolism. In addition, particularly significant responses could be identified in certain pathways for some conditions. Amino acid starvation induced a large number of genes in amino acid metabolism; large numbers of induced genes could also be observed in carbohydrate metabolism for heat shock and diamide treatment, and in glycan and energy metabolism for diauxic shift experiments. Large numbers of repressed genes could be identified in carbohydrate metabolism for alternative carbon source experiments, in lipid and amino acid metabolism for nitrogen depletion experiments, and in energy metabolism for hyper-osmotic shocks.

Table 1. $z_i^{(s)}$ scores statistics for different conditions

Condition	Average of $z_i^{(s)}$	Standard deviation of $z_i^{(s)}$
Nitrogen depletion (N)	−0.278	1.019
Stationary phase (S)	−0.100	1.071
Hyper-osmotic shock (Hr)	−0.014	1.027
Hydrogen peroxide treatment (H2)	−0.009	0.969
Diauxic shift (Dx)	0.035	0.982
Alternative carbon sources (AC)	0.057	1.065
Menadione exposure (M)	0.058	1.004
37°C heat shock (HS)	0.084	1.016
Dithiothrietol exposure (DTT)	0.084	0.978
Hypo-osmotic shock (Ho)	0.093	0.972
Diamide treatment (Dm)	0.100	1.002
Amino Acid starvation (AA)	0.153	1.273
Variable temperature shocks (VT)	0.200	0.879

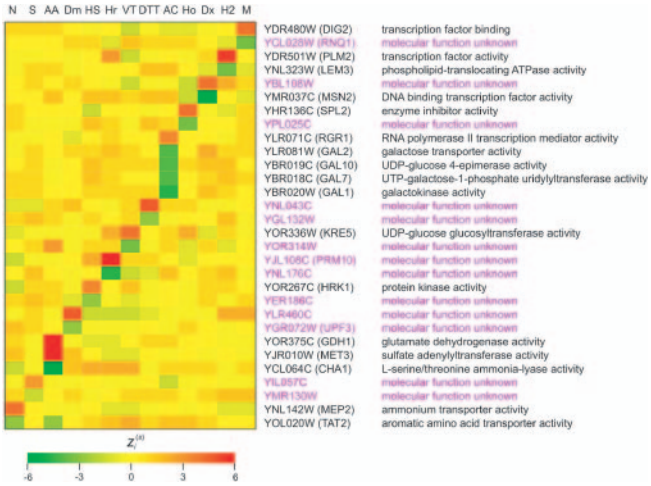


Fig. 2. Genes with maximum and minimum $z_i^{(s)}$ scores in each condition. Each row corresponds to one gene, with functional information from the *Saccharomyces* Genome Database when known. Each column corresponds to one specific condition, abbreviated as indicated in Table 1

We focused on those genes with the maximum (respectively minimum) $z_i^{(s)}$ scores in each condition. Not only do these genes exhibit a strong induction (repression) in a given condition, but this maximum (minimum) is furthermore unique to that condition (Figure 2). The functions of these genes are therefore expected to be closely linked to these particular conditions. For example, several GAL structural genes were identified among the most strongly correlated to alternative carbon source experiments, and it is known that these genes enable cells to utilize galactose as a carbon source (Lohr *et al*., 1995). For genes whose molecular role is unknown, functional information can thus be inferred by this approach. Interestingly, almost half on the genes identified by this analysis have no assigned molecular function in the *Saccharomyces* Genome Database (http://www.yeastgenome.org/).

In addition, we analyzed the relations between the expression responses to different external conditions. To quantify these relations, we computed the mutual correlation $I(s_1, s_2)$ of all pairs of conditions, defined as the Pearson coefficient correlation between two distributions of z scores $z_i^{(s1)}$ and $z_i^{(s2)}$. This analysis revealed that the expression responses to some conditions are significantly correlated (Figure 3a, b): strongly related pairs include alternative carbon sources and hypo-osmotic shock, nitrogen depletion and stationary phase, diamide treatment and heat shock, and hydrogen peroxide treatment and menadione exposure. At a larger scale, conditions can be grouped into five main clusters (see Table 1 for abbreviations): [N, S, AA], [Dm, HS], [Hr, VT, DTT, AC, Ho], [Dx], [H2, M]. The different types of couplings are clearly rendered by plotting the density distributions of pairs of $z_i^{(s)}$ scores. Examples for strongly positively correlated, uncorrelated, and anti-correlated pairs are shown in Figure 3c, d, e, respectively. It is worth noting that some of the responses are clearly uncoupled or anti-correlated with each other, in particular the first and third cluster described above. This correlation-based analysis thus reveals that cells can respond to different types of environmental shocks by significantly distinct patterns.

2.2 Signal-coupled gene sets build highly connected subnetworks

In order to understand to which extent correlations in expression patterns are linked to genes being involved in the same metabolic functions, expression data need to be integrated with topological modeling of metabolism. We reconstructed a network of genes involved in yeast metabolism from the KEGG database (see Methods). By considering only the genes with $z_i^{(s)} > 1$ (respectively $z_i^{(s)} < -1$), it is possible to construct subnetworks containing only the most significantly induced (repressed) genes in each particular condition (Figure 4).

These sets of significantly induced (repressed) genes were found to build remarkably well-connected subnetworks. In order to assess this point on an analytical basis, an extensive analysis on the topology of these induced (repressed) subnetworks was conducted. Several properties of the induced (repressed) subnetworks obtained under 13 different external conditions (*i.e*. a total number of 26 networks) were analyzed. The properties of the subnetworks were compared with those of the full metabolic network, as well as with random networks and artificially generated scale-free networks.

The normalized average path length in induced (repressed) subnetworks is in the same range as for random networks and other natural and artificial networks (Figure 5a). All signal induced (repressed) subnetworks show relative values inside the interval from 0.5 to 2, indicating that they have similar average shortest-path lengths as random networks. This behavior has been observed in many other networks, including the World Wide Web, protein interaction networks of yeast, collaboration networks of movie actors, etc (Dorogovtsev and Mendes, 2003; Barabási and Oltvai, 2004; van Noort *et al*., 2004).

However, signal induced (repressed) subnetworks show a higher connectivity than would be expected in random networks. This property is highlighted by plotting the relative values of the average clustering coefficients normalized by the average degrees (Figure 5b). The clustering value in the complete network is close to that of artificial scale-free networks constructed by the Barabási-Albert model (Barabási and Albert, 1999). Signal induced (repressed) subnetworks also appear in the vicinity of the scale-free model, but most of them show higher clustering coefficients. They are therefore more densely connected than scale-free networks. The clustering values in these subnetworks are also significantly higher than for random networks of equivalent size. This high connectivity is crucial for tracing back the sets of induced (repressed) genes to connected metabolic routes.

Two examples of such metabolic pathways are shown in Figure. 6 The red graph shows highly activated reactions ($z_i^{(s)} > 1$) when yeast cells are grown in minimal medium lacking amino acids. Interestingly, several amino acid producing pathways are entirely activated (for example, leucine, valine and lysine appear as final products in the bottom part of the figure). The green graph shows highly suppressed reactions ($z_i^{(s)} < -1$) when cells are grown in medium supplemented with glucose, galactose, raffinose, fructose, sucrose or ethanol as a carbon source. It is worth noting that glycolysis and the Krebs cycle are suppressed to a large extent.

The high connectivity of induced (repressed) subnetworks can be understood more widely as deriving from coexpression properties of genes controlling neighboring metabolic reactions. For

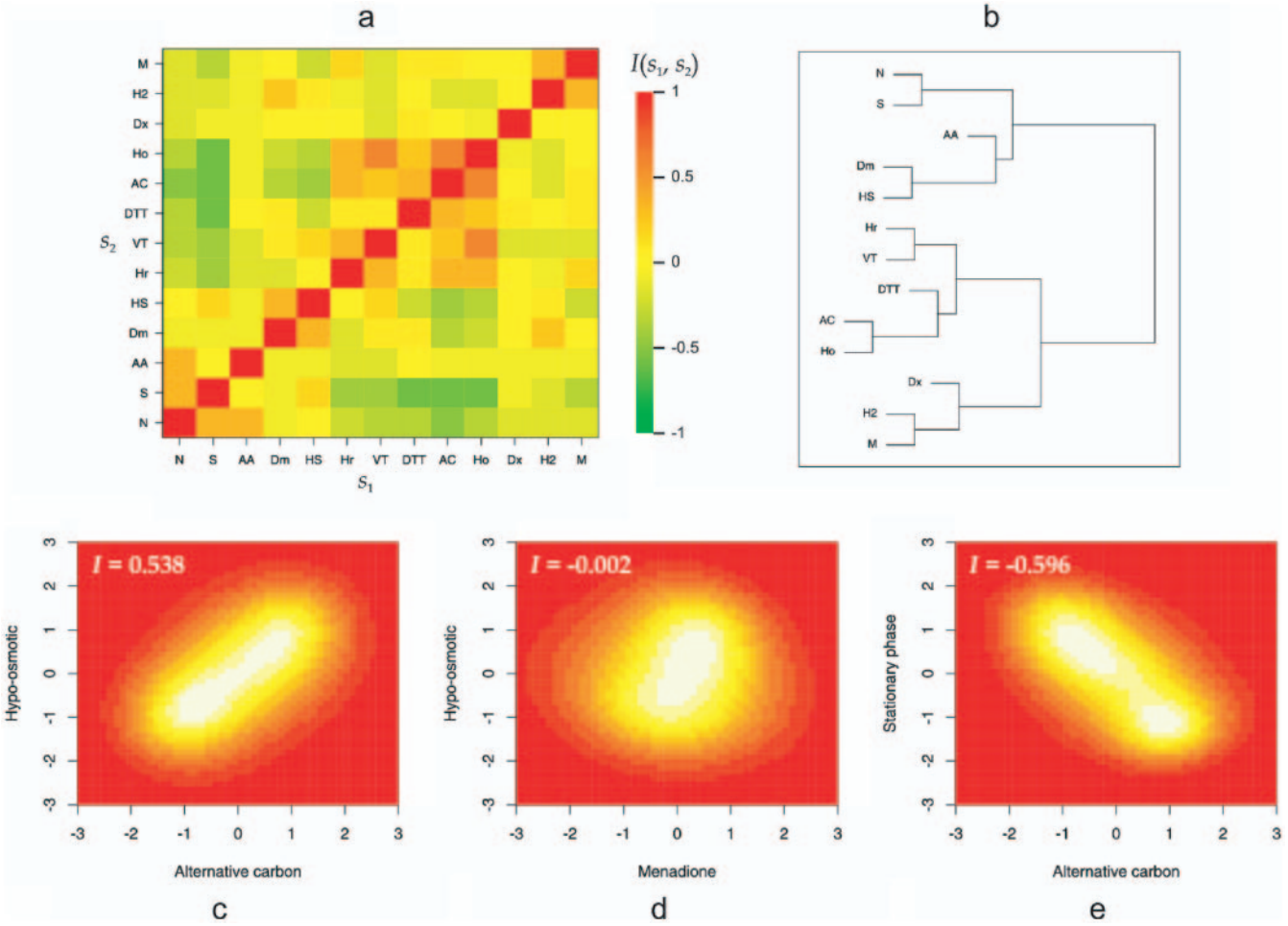


Fig. 3. (**a**) Mutual correlation $I(s_1, s_2)$ of all pairs of conditions. (**b**) Clustering dendogram of all conditions based on I values. Representative examples of density distributions of pairs of $z_i^{(s)}$ scores for strongly positively correlated (**c**), uncorrelated (**d**), and anti-correlated pairs (**e**).

each metabolite, we computed the average correlation between expression patterns of all its adjacent genes over all available experimental data (Figure 7). The distribution of correlation values with respect to node degree revealed a significant shift towards positive correlations. In order to verify whether this shift is statistically significant, we plotted the bounds of the interval computed from Fisher's transformation for a 99.7% confidence level centered on zero (see Methods). A large amount of compounds appear beyond this interval, including all of those with node degree higher than 10. This finding means that genes that are connected in the metabolic network tend to have correlated expression patterns at the genome scale: it is therefore expected that such local correlation leads induced (repressed) genes to be organized into connected subnetworks.

3 DISCUSSION

Two main findings have been derived from this analysis. Firstly, a correlation-based analysis of expression patterns in several different stress conditions allowed us to identify metabolic units whose activity is strongly coupled to a specific condition. Although the expression responses in a few groups of conditions showed positive correlations, most of these responses were found to be uncoupled with each other. Cells therefore seem to respond to different types of environmental shocks by more asymmetric patterns than described earlier (Gasch *et al*., 2000). Furthermore, several genes whose expression pattern is characterized by a strong and unique induction (repression) in a particular condition have been identified by this approach. The functions of these genes are expected to be closely linked to these particular conditions. The fact that several of these genes have not been assigned a molecular function previously indicates that this correlation-based approach can lead to novel insights about the role of genetic units.

Second, the integration of this correlation-based analysis to structural metabolic network data revealed that the sets of genes that are induced (repressed) under specific stress conditions define highly-connected subnetworks. This high connectivity is crucial for mapping such gene sets to precise metabolic routes. It should be noted that the subnetworks obtained by this approach have no relation with previously described ''gene networks''. The networks analyzed in this study are not built by linking genes together based on *a priori* information about some interactions or similarities, but

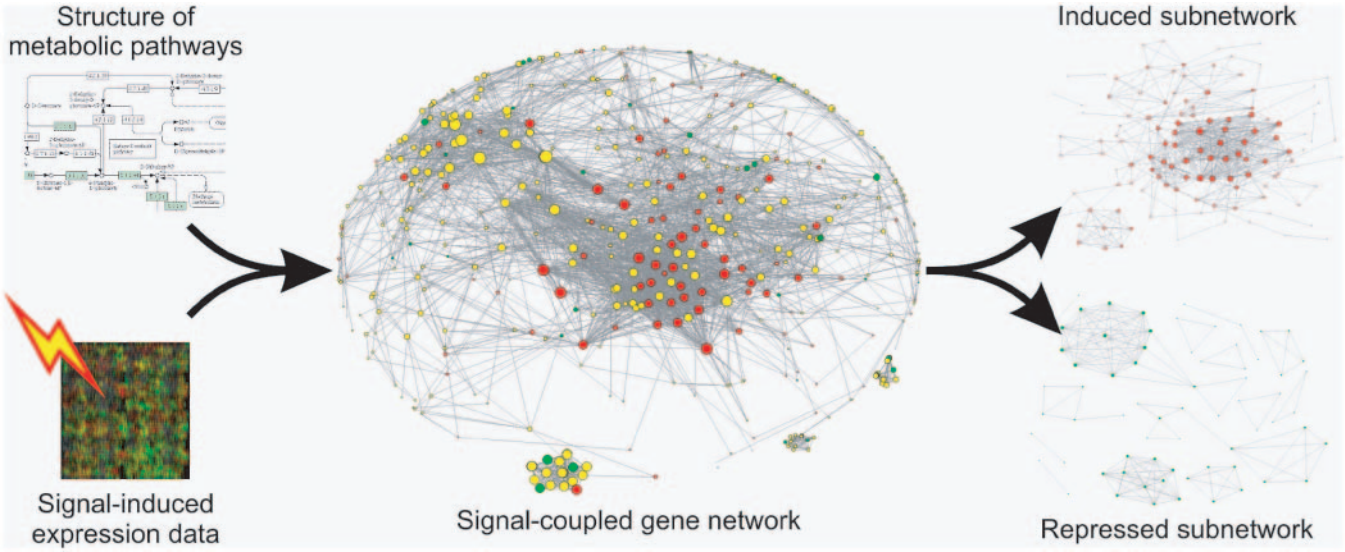


Fig. 4. Description of the integrative approach followed in this work. The architecture of the yeast metabolic pathways is combined with signal-induced gene expression data to produce gene networks coupled to each specific external signal. Each gene encoding for an enzyme is represented as a node, and two nodes are connected by an edge if the chemical reactions they control share at least one common chemical compound. From these networks, sets of significantly induced (repressed) genes are extracted. These genes define connected subnetworks whose topological properties can be analyzed. The networks shown in this figure are the ones obtained for amino acid starvation. The radii of nodes are proportional to the number of connections of each node. Colors have the same meaning as in Fig. 1

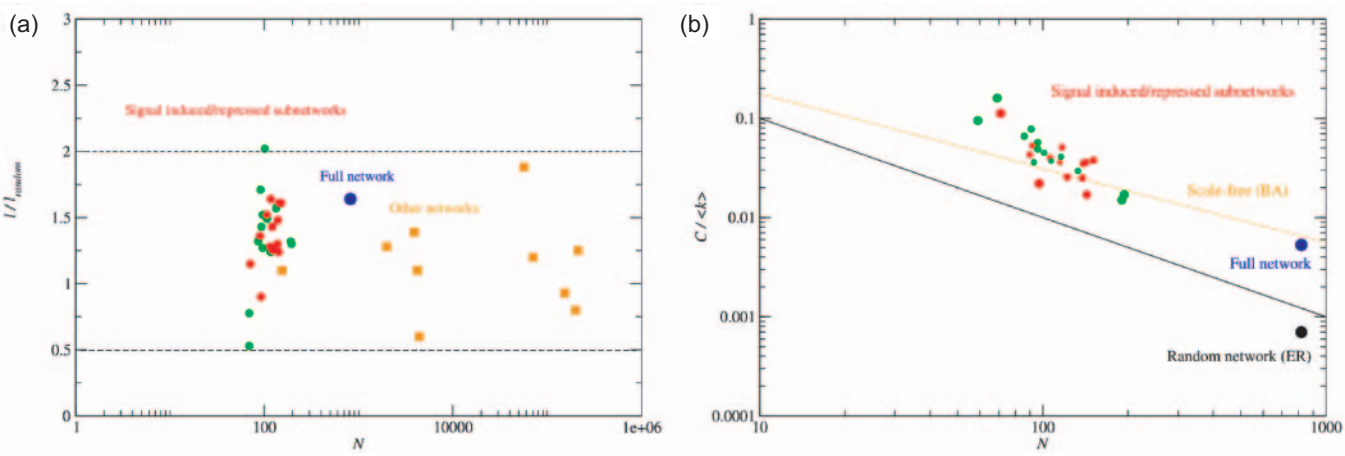


Fig. 5. (**a**) Ratio between the average shortest-path lengths of studied subnetworks and that of equivalent random networks, versus the network size *N*. Squares indicate the ratios of scale-free networks in other biological and non-biological systems. The blue circle indicates the full gene network. Red and green circles indicate the ratios for the signal-induced (repressed) subnetworks respectively. (**b**) Log-log plot of relative values of average clustering coefficients with respect to the average degree <*k*>, versus the network size *N*. The continuous line indicates the relation for a classical random graph. The dashed line indicates the behavior of scale-free networks. The blue circle indicates the relative clustering value for the full gene network. Red and green circles represent the values for the induced (repressed) subnetworks respectively. The high clustering values of induced (repressed) subnetworks reveal their high connectivity.

they are solely based on neighborhood of the gene products in the metabolic network. The aim of this study was indeed not to study the topological properties of known gene networks, but to find out to what extent genes which act in connected parts of the metabolic network exhibit coexpressed patterns. Although coexpression of genes controlling connected metabolic reactions has been observed in small metabolic subunits (DeRisi *et al*., 1997; Ihmels *et al*., 2004), it has never been observed at the genome scale to our knowledge.

Our topological analysis revealed the compactness of gene networks which regulate chemical reactions in active pathways. Compactness emerges in many real networks when a small average shortest-path is combined with a relatively high clustering coefficient. In this situation, a network is said to have the

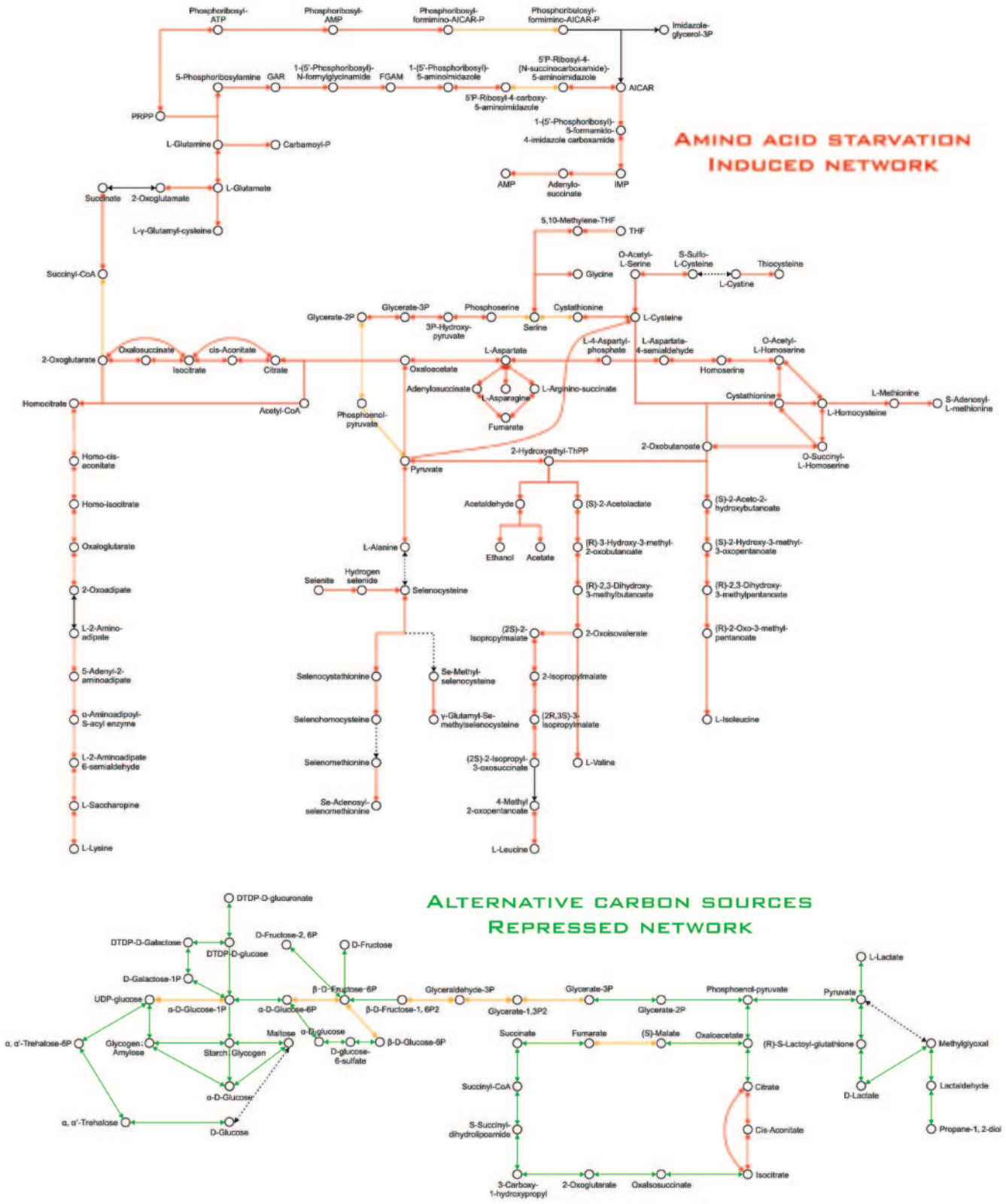


Fig. 6. Two examples of significantly induced (repressed) metabolic pathways. These connected metabolic routes can be reconstructed thanks to the high connectivity of the signal-coupled gene subnetworks. Colors have the same meaning as in Figure 1.

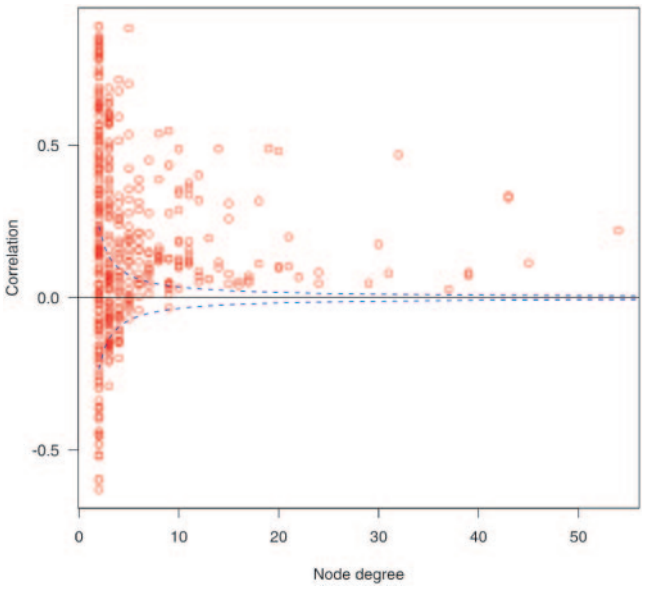


Fig. 7. Average correlation between expression patterns of all adjacent genes of each chemical compound over all available experimental data, with respect to node degree. The dash lines indicate the bounds of the 99.7% confidence interval computed from Fisher's transformation.

small-world property (Watts *et al*., 1998). In small-world networks, clustering coefficients are much higher than in the corresponding random networks. In contrast, the shortest-path lengths tend to be close to those of random graphs.

Here, the full gene network is characterized by a high clustering coefficient and by an average shortest-path length compatible with random graph networks. Furthermore, the subnetworks built by significantly induced (repressed) genes in particular external conditions show similar topological properties to the complete network. This effect was observed for all the external conditions we analyzed.

Recent analyses revealed that it is not possible to draw conclusions on the topological properties of subnetworks from the original properties of the full network (Stumpf *et al*., 2005; Han *et al*., 2005). Therefore, the compactness of induced (repressed) subnetworks could not be inferred without conducting a topological analysis.

The approach presented in this work is simple and fast, and could be easily applied to higher organisms or plants where the number of genes is two or three times larger than in *S. cerevisiae*. An extension of this approach to higher organisms could be particularly useful for comparing the metabolic responses of normal and diseased cells, and elucidating those metabolic functions which are disturbed by specific pathologies or affected by specific drugs. Furthermore, metabolites that are connected to highly induced (repressed) genes are likely to be very affected themselves by the specific perturbation and therefore constitute interesting potential targets for drugs.

4 CONCLUSION

The finding that induced (repressed) genes are highly connected in metabolic pathways raises the question of what mechanisms regulate this collective response to environmental stress. These mechanisms may be investigated by embedding the transcriptional regulatory networks into the current framework. Although this integration is not straightforward, it would provide valuable insights into the functionalities of genes and transcriptional factors involved in responses to external signals. Additional future progress would be required in the development of automatic procedures for identifying sets of genes connected in metabolism, mapping these genes back to biochemical pathways, and visualizing the induced (repressed) metabolic backbones.

5 METHODS

5.1 Datasets

The structural data for the metabolic network of *S. cerevisiae* were obtained from the specialized section of the KEGG database. This network contains 12456 chemical compounds and 6534 chemical reactions. Genes encoding the enzymes catalyzing chemical reactions can be identified in KEGG as well. Data of the whole-genome gene expression of *S. cerevisiae*, composed of 6152 genes, were downloaded from the site http://www-genome.standford.edu/yeast_stress. They are based on the experimental work of Gasch *et al*. (2000). From 173 experiments present in the original expression data, only 162 experiments making use of wild-type yeast were retained, and experiments using mutant strains were removed for the dataset. The yeast strain used in our study corresponds to DBY9434. DNA microarrays were used to analyze the changes in the amount of mRNA in yeast cells responding to 13 different environmental stresses. The yeast cells grew in rich medium at 30°C and were shaken at 250-300 rpm before the environmental shocks were applied. These 162 experiments were classified into 14 different sets, including one control set and 13 different shock sets: 37°C heat shock (HS), variable temperature shocks (VT), hydrogen peroxide treatment (H2), menadione exposure (M), diamide treatment (Dm), dithiothrietol exposure (DTT), hyper-osmotic shock (Hr), hypo-osmotic shock (Ho), amino acid starvation (AA), nitrogen depletion (N), diauxic shift (Dx), stationary phase (S), and alternative carbon sources (AC).

5.2 Correlation between expression data and specific conditions

Gene expression data for the different conditions were gathered in a matrix. Each row in the matrix corresponded to one gene of *S. cerevisiae* and each column corresponded to one environmental condition. Each environmental condition may be reproduced several times, the number of occurrences varying between 5 and 22. The total dimensions of the matrix were $N_g = 6152$ rows versus $N_c = 162$ columns.

In order to be able to compare data from different experiments, we renormalized the gene expression data so that the mean value of gene expression values was set to 0 and the standard deviation to 1. These rescaled values were obtained by subtracting for each gene i the average expression value μ_i calculated for N_c values and by dividing the resulting value by the standard deviation σ_i. The value of the expression level for each gene i in a specific entry t after normalization was denoted by $g_i(t)$, with $t = 1, 2, \ldots, N_c$:

$$g_i(t) = \frac{G_i(t) - \mu_i}{\sigma_i} \quad (1)$$

where $G_i(t)$ is the expression value of gene i in specific entry t before normalization.

For each specific stress condition s, an idealized expression pattern $g_s(t)$ with N_c dimensions was then defined, whose components took the binary values 1 or 0. For each condition s and for each entry t, vector $g_s(t)$ contained values of one if the stress condition was present in t, zero if not. Next, the

covariance between gene expression values $g_i(t)$ and the idealized pattern $g_s(t)$ was computed as follows:

$$\mathrm{cov}_i = \sum_{t=1}^{N_c} g_i(t) \cdot g_s(t) \quad (2)$$

Finally, for each gene i, $\mathrm{cov}_i^{(s)}$ was normalized by substracting the average $\mu_{\mathrm{cov}}^{(s)}$ and dividing by the standard deviation $\sigma_{\mathrm{cov}}^{(s)}$ of all genes. We thus defined for each gene i and each condition s:

$$z_i^{(s)} = \frac{\mathrm{cov}_i^{(s)} - \mu_{\mathrm{cov}}^{(s)}}{\sigma_{\mathrm{cov}}^{(s)}} \quad (3)$$

The value of $z_i^{(s)}$ measures the degree of coupling between the expression pattern of gene i and a given condition s. If gene i is strongly induced (repressed), $z_i^{(s)}$ has a high positive (negative) value. In particular, we considered in this study that genes with $z_i^{(s)}$ above (below) the value of 1 were significantly induced (repressed) in the given condition. Genes with $z_i^{(s)}$ values between -1 and 1 were considered to exhibit a weak coupling to the condition.

Related approaches have already been presented in different contexts, as for example medical classification (Golub *et al*., 1999), the identification of ploidy-regulated genes (Galitski *et al*., 1999), and the analysis of transcriptional regulatory networks (Balázsi *et al*., 2005).

Computations were performed using the R software environment (www.r-project.org) and custom-designed software.

5.3 Construction of gene network from metabolic structural data

In this study, the metabolism of *S. cerevisiae* was analyzed using the KEGG database, in which nodes represent chemical compounds and edges represent biochemical reactions. It is possible to define two complementary representations of a metabolic network (Wagner and Fell, 2001; Nacher *et al*., 2005). First, a *chemical network* consisting of nodes as chemical compounds can be created. In this representation, two nodes are connected by an edge if they are involved in the same chemical reaction. Second, it is also possible to construct a *gene network*, where nodes are the genes encoding for enzymes which catalyze the chemical reactions between compounds. In this network, two genes are connected by an edge if the reactions they control share at least one common chemical compound. If two or more genes control the same reaction, they are also connected by an edge. When the same gene is involved in several chemical reactions it is represented as a single node.

One drawback of the latter representation is that a large number of connections are created by ubiquitous metabolites (such as ATP, ADP...) but do not correspond to real metabolic flows (Ma and Zeng, 2003; Arita, 2004; Croes *et al*., 2005). In order to eliminate most of those connections, we cured our network by removing all connections created by the following list of ubiquitous metabolites: water, ATP, ADP, AMP, NADPH, NADP, NADH, NAD, CO_2, NH_3, O_2, H^+, orthophosphate, pyrophosphate, and CoA. The following list of compounds were removed too, because they are used as generic identifiers in KEGG and they can actually correspond to different chemical compounds in different reactions: protein, phosphoprotein, alcohol, and aldehyde.

For network visualization, the Pajek software was used (http://www.vlado.fmf.uni-lj.si/pub/networks/pajek).

5.4 Network analysis

A network consists of nodes connected by edges, and the number of connections to a node is called degree k. Characteristic properties of a network include the total number of nodes N, the average degree $\langle k \rangle$, and the degree distribution $P(k)$, which indicates the probability to find nodes with degree k. While in a random network the degree distribution has a peak close to the average value k (Erdős and Rényi, 1960), in scale-free networks there is a statistical abundance of nodes with high degree which generates a degree distribution with a power-law tail (Barabási and Albert, 1999). Interestingly, most of real networks are scale-free networks.

The *compactness* (or *small-worldness*) of real networks has recently captured the attention of the scientific community (Watts and Strogatz, 1998). Two concepts are useful for studying compactness properties: the average path length and the clustering coefficient. First, if we consider that the edges of a network have the same length and use it as a unit of length, then the distance between two nodes is the length of the shortest path between them. The distribution of the distances l between all reachable pairs of nodes, denoted by $P(l)$, indicates the probability that the length of the shortest path between two randomly chosen nodes is equal to l. In a random network, the average distance can be calculated by using the following expression: $l_{\mathrm{rand}} \sim \ln N / \ln \langle k \rangle$. In contrast, for a scale-free network, the expression reads: $l_{\mathrm{SF}} \sim \ln N / \ln(\ln N)$ (Dorogovtsev and Mendes, 2003).

Second, the clustering coefficient characterizes the density of edges in the neighborhood of a node. Given a node i with k neighbors, $C_i(k)$ denotes the probability that two nearest neighbors of node i are connected to each other, and takes values from 1 (fully connected network) to 0 (tree graph). Again, it is possible to derive the analytical expressions of the average clustering coefficient for random graphs and scale-free networks: $C_{\mathrm{rand}} = \langle k \rangle / N$ for a random graph, and $C_{\mathrm{SF}} \sim N^{-0.75}$ for the Barabási-Albert scale-free model (Albert and Barabási, 2002).

5.5 Fisher's transformation

Fisher's transformation is used for computing confidence intervals on Pearson's correlation between two variables. The formula for the transformation is:

$$\frac{1}{2} \ln\left(\frac{1+r}{1-r}\right) \pm \frac{q}{\sqrt{n-3}} = \frac{1}{2} \ln\left(\frac{1+\rho}{1-\rho}\right) \quad (4)$$

where n is the number of observations and q is the quantile of the chosen confidence interval (q = 3 for a 99.7% confidence). For a given value of r, the two values of ρ are the bounds of the confidence interval corresponding to a statistically significant correlation between the two variables. In our case r = 0, as the aim was to verify whether observed correlations are significant. Observations whose correlation values are outside the confidence interval can therefore be considered as significantly correlated, with a 99.7% level of confidence.

ACKNOWLEDGEMENTS

This work was supported by grants from the Ministry of Education, Culture, Sports, Science and Technology of Japan, the Japan Society for the Promotion of Science, the Japan Science and Technology Corporation, and by Grant-in-Aid ''Systems Genomics''. The computational resource was provided by the Bionformatics Center, Institute for Chemical Research, Kyoto University.

REFERENCES

Albert, R. and Barabási, A.-L., (2002) Statistical mechanics of complex networks. *Reviews of Modern Physics*, **74**, 47–97.

Almaas, E. et al. (2004) Global organization of metabolic fluxes in the bacterium *E. coli*. *Nature*, **427**, 839–843.

Arita, M. (2004) The metabolic world of E. coli is not small. *Proc. Natl. Acad. Sci. USA*, **101**, 1543–1547.

Balázsi, G. *et al*. (2005) Topological units of environmental signal processing in the transcriptional regulatory network of Escherichia coli. *Proc. Natl. Acad. Sci. USA*, **102**, 7841–7846.

Barabási, A.-L. and Oltvai, Z.N. (2004) Network biology: Understanding the cell's functional organization. *Nat. Rev. Genet*., **5**, 101113.

Barabási, A.-L. and Albert, R. (1999) Emergence of scaling in random networks. *Science*, **286**, 509–512.

Barriot, R. *et al*. (2004) New strategy for the representation and the integration of biomolecular knowledge at a cellular scale. *Nucl. Acids Res*., **32**, 3581–3589.

Borisjuk, L. *et al*. (2004) Integrating data from biological experiments into metabolic networks with the DBE information system. *In Silico Biology*, **5**, 11.

Croes, D. *et al*. (2005) Metabolic PathFinding: inferring relevant pathways in biochemical networks. *Nucl. Acids Res*., **33**, w326-w330.

Dahlquist, K.D. *et al*. (2002) GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nat. Genet*., **31**, 19–20.

DeRisi, J.L. *et al*. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **278**, 680–686.

Dorogovtsev, S.N. and Mendes, J.F.F. (2003) Evolution of Networks: From Biological Nets to the Internet and WWW, Oxford University Press, Oxford.

Edwards, J.S. and Palsson, B.Ø. (2000) The E. coli MG1655 in silico metabolic genotype: its definition, characteristics, and capabilities. *Proc. Natl. Acad. Sci. USA*, **97**, 5528–5533.

Erdős, P. and Rényi, A. (1960) On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*., **5**, 17–61.

Förster, J. *et al*. (2003) Genome-scale reconstruction of the S. cerevisiae metabolic network. *Genome Res*., **13**, 244–253.

Galitski, T. *et al*. (1999) Ploidy regulation of gene expression. *Science*., **285**, 251–254.

Gasch, A.P. and Werner-Washburne, M. (2002) The genomics of yeast responses to environmental stress and starvation. *Funct. Integr. Genomics*, **2**, 181–192.

Gasch, A.P. *et al*. (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cel*., **11**, 4231–4257.

Golub, T.R. *et al*. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*., **286**, 531–537.

Guelzim, N. *et al*. (2002) Topological and causal structure of the yeast transcriptional regulatory network. *Nat. Genet*., **31**, 60–63.

Han, J.-D. J. *et al*. (2005) Effect of sampling on topology predictions of protein-protein interaction networks. *Nat. Biotechnol*., **23**, 839–844.

Ideker, T. *et al*. (2001) Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*., **292**, 929–934.

Ihmels, J. *et al*. (2004) Principles of transcriptional control in the metabolic network of S. cerevisiae. *Nat. Biotechnol*., **22**, 86–92.

Jeong, H. *et al*. (2000) The large-scale organization of metabolic networks. *Nature*., **407**, 651–654.

Kanehisa, M. *et al*. (2006) From genomics to chemical genomics: new developments in KEGG. *Nucl. Acids Res.,* **34**, D354-D357.

Krömer, J.O. *et al*. (2004) In-depth profiling of lysine-producing Corynebacterium glutamicum by combined analysis of the transcriptome, metabolome, and fluxome. *J. Bacteriol*., **186**, 1769–1784.

Lee, J.S.M *et al*. (2005) GObar: A Gene Ontology based analysis and visualization tool for gene sets. *BMC Bioinformatics*, **6**, 189.

Lohr, D. *et al*. (1995) Transcriptional regulation in the yeast GAL gene family: a complex genetic network. *FASEB J*., **9**, 777–787.

Ma, H. and Zeng, A.-P. (2003) Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics*., **19**, 270–277.

Mlecnik, B. *et al*. (2005) PathwayExplorer: web service for visualizing high-throughput expression data on biological pathways. *Nucl. Acids Res*., **33**, W633-W637.

Nacher, J.C. *et al*. (2005) Two complementary representations of a scale-free network. *Physica A*, **349**, 349–363.

Patil, K.R. and Nielsen, J. (2005) Uncovering transcriptional regulation of metabolism by using metabolic network topology. *Proc. Natl. Acad. Sci. USA*, **102**, 2685–2689.

Schilling, C.H. *et al*. (2000) Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *J. Theor. Biol*., **203**, 229–248.

Schuster, S. *et al*. (2000) A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nat. Biotechnol*., **18**, 326–332.

Stumpf, M.P.H. *et al*. (2005) Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proc. Natl. Acad. Sci. USA*, **102**, 4221–4224.

Tian, L. *et al*. (2005) Discovering statistically significant pathways in expression profiling studies. *Proc. Natl. Acad. Sci. US*., **102**, 13544–13549.

van Noort, V. *et al*. (2004) The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model. *EMBO Rep*., **5**, 280–284.

Wagner, A. and Fell, D.A. (2001) The small world inside large metabolic networks. *Proc. R. Soc. London B*, **268**, 1803–1810.

Watts, D.J. and Strogatz, S.H. (1998) Collective dynamics of 'small-world' networks. *Nature*., **393**, 440–442.

Zaslaver, A. *et al*. (2004) Just-in-time transcription program in metabolic pathways. *Nat. Genet*., **36**, 486–491.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e384–e392
doi:10.1093/bioinformatics/btl251

Informative priors based on transcription factor structural class improve *de novo* motif discovery

Leelavati Narlikar[1,*], Raluca Gordân[1,*], Uwe Ohler[1,2,*] and Alexander J. Hartemink[1,2,*]

[1]Department of Computer Science, Duke University, Durham, NC 27708 and [2]Institute for Genome Sciences and Policy, Duke University, Durham, NC 27708.

ABSTRACT

Motivation: An important problem in molecular biology is to identify the locations at which a transcription factor (TF) binds to DNA, given a set of DNA sequences believed to be bound by that TF. In previous work, we showed that information in the DNA sequence of a binding site is sufficient to predict the structural class of the TF that binds it. In particular, this suggests that we can predict which locations in any DNA sequence are more likely to be bound by certain classes of TFs than others. Here, we argue that traditional methods for *de novo* motif finding can be significantly improved by adopting an informative prior probability that a TF binding site occurs at each sequence location. To demonstrate the utility of such an approach, we present PRIORITY, a powerful new *de novo* motif finding algorithm.

Results: Using data from TRANSFAC, we train three classifiers to recognize binding sites of basic leucine zipper, forkhead, and basic helix loop helix TFs. These classifiers are used to equip PRIORITY with three class-specific priors, in addition to a default prior to handle TFs of other classes. We apply PRIORITY and a number of popular motif finding programs to sets of yeast intergenic regions that are reported by ChIP-chip to be bound by particular TFs. PRIORITY identifies motifs the other methods fail to identify, and correctly predicts the structural class of the TF recognizing the identified binding sites.

Availability: Supplementary material and code can be found at http://www.cs.duke.edu/~amink/.

Contact: lee@cs.duke.edu, raluca@cs.duke.edu, uwe.ohler@duke.edu, amink@cs.duke.edu.

1 INTRODUCTION

Transcriptional regulation is governed in large part by interactions between DNA-binding proteins called transcription factors (TFs) and the corresponding sites on the DNA to which they bind. TF proteins have specific three-dimensional structures crucial for recognition of their binding sites. The binding affinity, and hence the transcription of the regulated gene, depends on both the TF's DNA-binding domain and the site it recognizes. A TF usually binds multiple sites sharing some common structure, which is typically represented using a statistical or word-based model.

An important problem in deciphering the gene regulatory code is to be able to find *de novo* binding sites for a TF given a collection of DNA sequences thought to be bound by that TF (Wasserman, 2004; Siggia, 2005). Recent advances in gene-expression arrays (Spellman *et al*., 1998; Kim *et al*., 2001, and many more), ChIP-chip experiments (Harbison *et al*., 2004; Liu *et al*., 2005), and *in vitro* DNA-binding arrays (Mukherjee *et al*., 2004) have resulted in an explosion of such data. Finding the most probable locations of binding sites hidden within the DNA sequences, and hence learning the motif best describing these binding sites, constitutes a problem of parameter estimation over an exponential search space.

Current motif finding algorithms commonly have difficulty when the motifs describing a set of binding sites are quite weak, in the sense that they are not especially over-represented relative to background. In such cases, additional information might be useful in guiding an algorithm to these weaker motifs, perhaps 'up-weighting' them relative to background so that they can be detected. This can be done using comparative genomic information, but even that information will not handle another common problem, illustrated by the following scenario. Imagine that TF_1 binds to a particular set of DNA sequences but that many of those same sequences are also bound by TF_2. If the motif of TF_2 is much stronger than that of TF_1, then the motif for TF_2 will be reported as the motif for both TFs, even if the TFs recognize and bind to DNA in quite different ways. In this paper, we present a way to overcome both of these problems.

Most eukaryotic TFs can be classified based on the structure of their DNA-binding domains. Due to the co-evolution of TFs with their binding sites, one might expect that just as TFs with a similar structure have similar DNA-binding mechanisms, there might be corresponding similarities within the DNA binding sites of TFs with similar DNA-binding mechanisms. Indeed, in a previous paper (Narlikar and Hartemink, 2006), we have shown that it is possible to predict the structural class of a TF using neither its amino acid sequence nor other protein structure information, but only the sequences of its DNA binding sites. Briefly, we built a multiclass classifier to distinguish between TFs of six different classes—Cys_2His_2 zinc fingers, Cys_4 zinc fingers, basic helix loop helix, basic leucine zippers, forkheads, and homeodomains—using only features of the sequences of their binding sites. We were able to correctly classify 87% of the TFs in a leave-one-out cross-validation procedure. Here, we build a set of binary classifiers which classify short DNA sequences as either binding sites of a particular structural class or not. We extract a large number of sequence features from these binding sites, and train a sparse Bayesian classifier based on logistic regression for this purpose. We adopt the output from three such classifiers as priors in Gibbs sampling to search for TF binding sites. The goal of these priors is for the search algorithm to be able to more rapidly and sensitively capture the ''true'' motif of the TF. This

*To whom correspondence should be addressed.

motif is expected to be based on the known binding properties of TFs sharing the same DNA-binding domain, and not just statistical over-representation relative to a background model of the sequence.

We show that our algorithm, called PRIORITY, is able to identify motifs that are not selected by popular motif finding algorithms. Along with the best motif, our algorithm outputs the most likely class to which the TF belongs. Also, when the class of the TF is known and a specific class prior can be applied by itself, we show that the resulting algorithm converges in significantly fewer iterations than when using a uniform prior. Our choice of Gibbs sampling over other search methods like expectation maximization (Dempster *et al*., 1977) is arbitrary; the concept of class-specific location priors can be applied in either context. Our choice of a position specific score matrix (PSSM), which stores the preference for each putative nucleotide at each position of the binding site (Staden, 1984), as a model for binding sites is also arbitrary; we use this model because it is widely used, and again, the concept of class-specific location priors can be incorporated with nearly any model of a TF binding site. The purpose of this paper is to show how using informative priors with respect to locations in the DNA sequences (here based on the TF structural class) improves motif discovery in general.

2 APPROACH

In this section we start with the description of the sequence model, go on to describe the generation of the class prior, and finally explain the Gibbs sampling strategy for the actual search.

2.1 Model framework

2.1.1 Sequence model Assume we have n DNA sequences $\boldsymbol{X}_1$ to $\boldsymbol{X}_n$ believed to be bound by the same TF. For simplicity, we assume that there is at most one instance of a binding site (or DNA motif) of that TF of length W hidden in each sequence (analogous to the zero or one occurrence per sequence model, or ZOOPS, in MEME (Bailey and Elkan, 1994)), though we can extend this approach to finding multiple instances of the binding site (analogous to the two component mixture model in MEME), as is implemented by Thijs *et al*. (2002). The motif follows a PSSM model while the rest of the sequence follows some pre-calculated background model $\boldsymbol{\phi}_0$. The PSSM can be described by a matrix $\boldsymbol{\phi}$ where $\phi_{a,b}$ is the probability of finding base b at location a within the binding site for $1 \leq b \leq 4$ and $1 \leq a \leq \mathrm{W}$. Let $\boldsymbol{Z}$ be a vector of size n denoting the starting location of the binding site in each sequence: $Z_i = j$ if there is a binding site starting at location j in $\boldsymbol{X}_i$ and we adopt the convention that $Z_i = 0$ if there is no binding site in $\boldsymbol{X}_i$. Thus if the sequence $\boldsymbol{X}_i$ is of length m_i and if $\boldsymbol{X}_i$ contains a binding site at location Z_i, we can compute the probability of the sequence given the model parameters as:

$$P(\boldsymbol{X}_i \mid \boldsymbol{\phi}, Z_i > 0, \boldsymbol{\phi}_0) = (X_{i,1}, X_{i,2}, \ldots, X_{i,Z_i-1} \mid \boldsymbol{\phi}_0) \times \prod_{k=Z_i}^{Z_i+W-1} \phi_{k-Z_i+1, X_{i,k}} \times P(X_{i,Z_i+W}, \ldots, X_{i,m_i} \mid \boldsymbol{\phi}_0)$$

and if it does not contain a binding site as:

$$P(\boldsymbol{X}_i \mid \boldsymbol{\phi}, Z_i = 0, \boldsymbol{\phi}_0) = P(X_{i,1}, X_{i,2} \ldots X_{i,m_i} \mid \boldsymbol{\phi}_0)$$

2.1.2 Objective function We wish to find $\boldsymbol{\phi}$ and $\mathbf{Z}$ to maximize the joint posterior distribution of all the unknowns given the data. Hence, the objective function is:

$$\arg\max_{\boldsymbol{\phi}, \boldsymbol{Z}} \; P(\boldsymbol{\phi}, \boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\phi}_0) \tag{1}$$

2.2 Calculation of the prior

Most motif discovery algorithms assume *a priori* that a binding site is uniformly likely to occur in all locations within each sequence. However, since we have demonstrated that certain sequences are more or less likely to be bound by various classes of TFs, we can build an informative prior to reflect such an *a priori* bias. To do so, we create three binary classifiers. The first one classifies a DNA subsequence as a binding site of a basic leucine zipper (bZip) TF or not a binding site of a bZip TF. The second distinguishes between forkhead binding sites and forkhead non-binding sites. The third distinguishes between basic helix loop helix (bHLH) binding sites and bHLH non-binding sites.

To build training sets for these classifiers, we use binding sites listed in TRANSFAC 9.4 (Wingender *et al*., 2001) that fall into one of these classes. We remove binding sites belonging to *Saccharomyces cerevisiae* from this set, since we intend to test the algorithm on yeast TFs. This leaves us with 1131 bZip, 466 forkhead, and 325 bHLH binding sites. For the training set of non-binding sites, we use a third-order Markov model from yeast intergenic regions and randomly sample subsequences of the same length distribution as the binding sites from that Markov model. We include three times as many non-binding sites as binding sites for each classifier to provide enough coverage.

For each sequence in the three training sets we construct a vector of length 1387 describing possibly relevant features of this sequence. These sequence features include:

(1) *Subsequence frequency features (1364)*: Integers representing counts of all subsequences of length 1 (i.e., each of the four nucleotides) to length 5 (i.e., each of the 4^5 possible nucleotide strings). These integers account for a total of 1364 entries in the vector, comprising the vast majority of possibly relevant features.

(2) *Ungapped palindrome features (8)*: Binary indicator variables denoting whether the sequence contains palindromic[1] subsequences of half-length 3, 4, 5, or 6 that span the entire site (i.e., end to end), as well as those that do not span the entire site (i.e., are somewhere in the middle of the site).

(3) *Gapped palindrome features (8)*: Binary indicator variables denoting whether the sequence contains gapped palindromic subsequences of half-length 3, 4, 5, or 6 that span the entire site (i.e., end to end), as well as those that do not span the entire site (i.e., are somewhere in the middle of the site). A gapped palindromic subsequence is one in which some non-palindromic nucleotides are inserted exactly in the middle of two otherwise palindromic halves.

(4) *Special features (7)*: Binary indicator variables that denote the presence or absence of features that have been identified in the literature to be over-represented in the binding sites of certain classes of TFs.

[1]Throughout, we mean palindromic in the reverse complement sense.

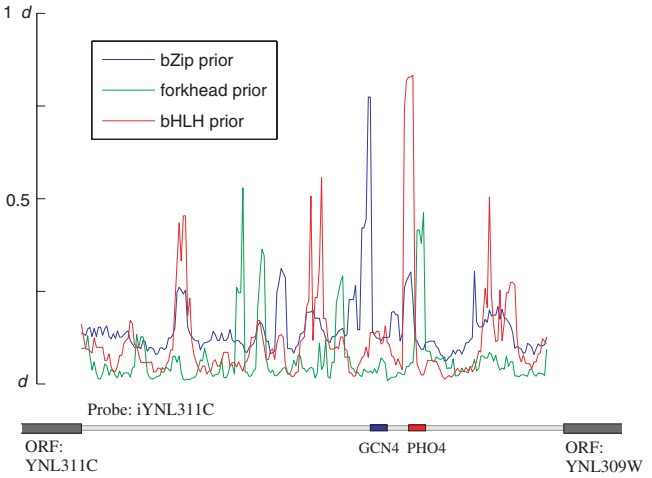


Fig. 1. Prior distributions for three classes on intergenic region iYNL311C in yeast. The *Y*-axis shows the C_{ijk} value ranging from d to $1-d$ (see text) for each of the three classes: bZip, forkhead, and bHLH where $\boldsymbol{X}_i$ is the sequence of the probe corresponding to iYNL311C. The blue and red boxes are putative motifs for Gcn4 and Pho4, respectively, predicted by Harbison *et al*. (2004) with the criterion of a probe for an intergenic region being bound with *p*-value < 0.001. Gcn4 is a bZip protein and Pho4 is a bHLH protein. As can be seen, the probabilities at the starting locations of these motifs are higher for the respective priors.

The classifiers are learned using Bayesian sparse multinomial logistic regression (SMLR), which is designed to select a small set of features relevant for classification (Krishnapuram *et al*., 2005). The fact that features in binding sites can be used to predict the structure of the DNA-binding domain of a TF has been shown by Narlikar and Hartemink (2006) where a six-way classifier was built based on the same DNA sequence features to distinguish between TFs belonging to one of six different structural classes. We estimate the generalization accuracy using 10-fold cross-validation and achieve 89.6%, 95.2%, and 95.1% for the bZip, forkhead, and bHLH binary classifiers respectively.

Each binary classifier, being based on logistic regression, outputs the probability of the input sequence being a binding site of the respective class. Since the classifiers have a nonzero misclassification rate, instead of using the probabilities reported by the classifier directly, we linearly scale them to lie in the interval $[d, 1-d]$, where $0 \leq d \leq 0.5$ is a tunable parameter. One can think of this transformation as a result of mixing with a uniform prior to dilute the effect of the classifier-based prior to a certain extent. Setting d to zero would be a special case in which the probabilities from the classifier are used as they are setting d to 0.5 would be a special case in which the probabilities from the classifier are ignored and a uniform prior is used instead. In all our analyses, we arbitrarily set d to 0.3.

In the general case in which r structural classes are modeled, the transformed output of the r classifiers is stored as a three dimensional vector $\boldsymbol{C}$ where C_{ijk} is the probability of the subsequence of length W starting at location j in sequence $\boldsymbol{X}_i$ being a binding site of class k and $(1-C_{ijk})$ is the probability of it not being a binding site of that class. For C_{ij0} (the probability of the subsequence being a binding site of a TF which is not a member of the r classes for which we have built classifiers), we use a uniform probability which can be an input from the user. In all our analyses, we arbitrarily set it to 0.4.

As an illustration, Figure 1 shows the values of C_{ijk} for the classes bZip, forkhead, and bHLH ($r = 3$), where $\boldsymbol{X}_i$ is the intergenic region iYNL311C in yeast. Also shown are the putative binding sites predicted by Harbison *et al*. (2004) when they use that region as a probe. As is evident from the figure, certain positions in the sequence are *a priori* more likely to contain a binding site of a particular class than others. The idea is to have such a prior distribution over locations in each sequence in $\boldsymbol{X}$ to aid motif discovery.

We now introduce $\boldsymbol{c}$, a vector of length n, where each c_i is a hidden variable representing the class of the TF that recognizes the binding site starting at Z_i in sequence $\boldsymbol{X}_i$. Each c_i can take a value from 1 to r representing the r classes or 0 to handle the possibility that the binding site belongs to none of the r classes. This allows us to robustly find motifs of TFs with totally different DNA-binding domains from those we model. We use another parameter $\boldsymbol{\gamma}$, a vector of length $r+1$ to define the multinomial parameters of $\boldsymbol{c}$.

Using $\boldsymbol{C}$ and $\boldsymbol{c}$, the prior probability on $\boldsymbol{Z}$ can be calculated as:

$$P(Z_i = 0 \mid c_i = k) \propto \prod_{j=1}^{m_i} (1 - C_{ijk}) \tag{2}$$

and for $u > 0$ as

$$P(Z_i = u \mid c_i = k) \propto C_{iuk} \prod_{\substack{j=1 \\ j \neq u}}^{m_i} (1 - C_{ijk}) \tag{3}$$

$P(Z_i \mid c_i)$ is normalized assuming the same proportionality constant in equations (2) and (3), so that under the assumptions of the model, we have

$$\sum_{j=0}^{m_i} P(Z_i = j \mid c_i = k) = 1 \quad \text{for } 0 \leq k \leq r$$

The inclusion of parameters $\boldsymbol{c}$ and $\boldsymbol{\gamma}$ changes the objective function in equation (1) to:

$$\arg\max_{\boldsymbol{\phi}, \boldsymbol{Z}, \boldsymbol{\gamma}, \boldsymbol{c}} P(\boldsymbol{\phi}, \boldsymbol{Z}, \boldsymbol{\gamma}, \boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\phi}_0) \tag{4}$$

2.3 Gibbs sampling

Gibbs sampling is a Markov chain Monte Carlo (MCMC) method that approximates sampling from a joint posterior distribution by sampling iteratively from individual conditional distributions (Gelfand and Smith, 1990). Let J_v denote the distribution function of parameter v conditional on the current values of all other parameters and data. We thus need to iteratively sample v from J_v for all unknown parameters v.

Applying the collapsed Gibbs sampling strategy developed by Liu (1994) for a faster convergence, we can integrate out both the $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$ and sample only the Z_i and c_i.

The expression for sampling $\boldsymbol{Z}$ from its conditional distribution is:

$$\begin{aligned} J_Z &= P(\boldsymbol{Z} \mid \boldsymbol{c}, \boldsymbol{X}, \boldsymbol{\phi}_0) \\ &\propto P(\boldsymbol{Z}, \boldsymbol{c}, \boldsymbol{X} \mid \boldsymbol{\phi}_0) \\ &= \int_{\boldsymbol{\gamma}, \boldsymbol{\phi}} P(\boldsymbol{\phi}, \boldsymbol{Z}, \boldsymbol{\gamma}, \boldsymbol{c}, \boldsymbol{X} \mid \boldsymbol{\phi}_0) \mathrm{d}\boldsymbol{\phi} \mathrm{d}\boldsymbol{\gamma} \\ &\propto P(\boldsymbol{Z} \mid \boldsymbol{c}) \int_{\phi} P(\boldsymbol{X} \mid \boldsymbol{\phi}, \boldsymbol{Z}, \boldsymbol{\phi}_0) P(\boldsymbol{\phi}) \mathrm{d}\boldsymbol{\phi} \end{aligned} \tag{5}$$

We get the above simplification since $\boldsymbol{Z}$ is independent of $\boldsymbol{\gamma}$ conditional on $\boldsymbol{c}$. By definition, the prior on $\boldsymbol{Z}$ is also independent of $\boldsymbol{\phi}$.

Similarly, $\boldsymbol{c}$ is independent of $\boldsymbol{\phi}$ and $\boldsymbol{\phi}_0$ conditional on $\boldsymbol{Z}$. We thus get an expression for sampling $\boldsymbol{c}$ from its conditional distribution:

$$\begin{aligned} J_{\boldsymbol{c}} &= P(\boldsymbol{c} \mid \boldsymbol{Z}, \boldsymbol{X}, \boldsymbol{\phi}_0) \\ &\propto P(\boldsymbol{Z}, \boldsymbol{c}, \boldsymbol{X} \mid \boldsymbol{\phi}_0) \\ &= \int_{\boldsymbol{\gamma}, \boldsymbol{\phi}} P(\boldsymbol{\phi}, \boldsymbol{Z}, \boldsymbol{\gamma}, \boldsymbol{c}, \boldsymbol{X} \mid \boldsymbol{\phi}_0) \mathrm{d}\boldsymbol{\phi} \mathrm{d}\boldsymbol{\gamma} \\ &= \int_{\boldsymbol{\gamma}} P(\boldsymbol{Z}, \boldsymbol{\gamma}, \boldsymbol{c}) \mathrm{d}\boldsymbol{\gamma} \int_{\boldsymbol{\phi}} P(\boldsymbol{X} \mid \boldsymbol{\phi}, \boldsymbol{Z}, \boldsymbol{\phi}_0) \, P(\boldsymbol{\phi}) \mathrm{d}\boldsymbol{\phi} \\ &\propto P(\boldsymbol{Z} \mid \boldsymbol{c}) \int_{\boldsymbol{\gamma}} P(\boldsymbol{c} \mid \boldsymbol{\gamma}) \, P(\boldsymbol{\gamma}) \mathrm{d}\boldsymbol{\gamma} \end{aligned} \tag{6}$$

Proceeding analogously to the derivation of Liu (1994), we can simplify the integrals using Dirichlet priors on both $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$. We derive the sampling distribution for Z_i, i.e J_{Z_i}, by computing $J_Z / J_{Z_{[-i]}}$ using equation (5), where $\boldsymbol{Z}_{[-i]}$ is the vector $\boldsymbol{Z}$ without Z_i. We further simplify the result by dividing it by $P(Z_i = 0, \boldsymbol{X}_i \mid c_i, \boldsymbol{\phi}_0)$ which is a constant at a particular sampling step. We thus have a sampling distribution for Z_i similar to the predictive update formula as described in Liu *et al*. (1995), but with the inclusion of the class prior:

$$J_{[Z_i = j]} = \frac{P(Z_i = j \mid c_i) \times \left(\prod_{a=1}^{W} \phi_{a, X_{i, j+a-1}} \right)}{P(Z_i = 0 \mid c_i) \times P(X_{i,j}, \ldots, X_{i, j+W-1} \mid \boldsymbol{\phi}_0)}$$

for $j > 0$, and

$$J_{[Z_i = j]} = 1$$

for $j = 0$ where $\boldsymbol{\phi}$ is calculated from the counts of the sites contributing to the current alignment $\boldsymbol{Z}_{[-i]}$ and the pseudocounts as determined by the Dirichlet prior.

Similarly, we get a sampling distribution for c_i:

$$\begin{aligned} J_{c_i} &= P(c_i \mid \mathbf{Z}, \mathbf{c}_{[-i]}) \\ &\propto P(Z_i \mid c_i = k) \times \gamma_k \quad \text{for } 0 \leq k \leq r \end{aligned}$$

where $\boldsymbol{\gamma}$ is calculated from the counts for each class from the current $\boldsymbol{c}_{[-i]}$ and the pseudocounts from the respective Dirichlet prior for $\boldsymbol{\gamma}$, where $\boldsymbol{c}_{[-i]}$ is the vector $\boldsymbol{c}$ without c_i.

We also provide the option of searching in the reverse complement of each sequence. This does not make a difference to any of the derivations. We simply concatenate the reverse complement of each $\boldsymbol{X}_i$ at the end of the original $\boldsymbol{X}_i$, and now the algorithm searches for zero or one occurrence of the motif in this longer sequence. Special care is taken to ensure that invalid locations (such as those spanning the concatenation boundary) have zero probability density during the sampling.

2.4 Scoring scheme

The joint posterior distribution function after each iteration can be calculated as:

$$\begin{aligned} P(\boldsymbol{\phi}, \boldsymbol{Z}, \boldsymbol{\gamma}, \boldsymbol{c} \mid \boldsymbol{X}, \boldsymbol{\phi}_0) \propto P(\boldsymbol{X} \mid \boldsymbol{\phi}, \boldsymbol{Z}, \boldsymbol{\phi}_0) \times P(\boldsymbol{Z} \mid \boldsymbol{c}) \\ \times P(\boldsymbol{c} \mid \boldsymbol{\gamma}) \times P(\boldsymbol{\phi}) \times P(\boldsymbol{\gamma}) \end{aligned} \tag{7}$$

To simplify the computation, we divide equation (7) by the constant probability $P(\boldsymbol{X} \mid \boldsymbol{Z} = \mathbf{0}, \boldsymbol{\phi}_0)$ and use the logarithm of the resulting function to score a motif.

In order to maximize the objective function and hence the score, we run the Gibbs sampler for a predetermined number of iterations after apparent convergence to the joint posterior, and output the highest scoring PSSM at the end.

3 RESULTS

We examined the ChIP-chip data published by Harbison *et al*. (2004) where the intergenic binding locations of TFs in yeast are profiled under various environmental conditions. We study the set of intergenic regions (or probes) that are bound with p-value < 0.001 by TFs belonging to one of the three classes for which we have built binary classifiers. There are a total of 24 TFs which qualify according to classification information in TRANSFAC, with a distribution of fourteen bZip, three forkhead, and seven bHLH proteins. We also use six more TFs whose binding sites have been well characterized in the literature, but do not fall in any of the three classes. This set is used to determine if our algorithm correctly learns motifs belonging to TFs in other structural classes for which we have not designed a specific binary classifier.

We compare the motifs found by our method to those found by Harbison *et al*. (2004). Harbison *et al*. use six different popular motif discovery programs: AlignACE (Roth *et al*., 1998), MEME (Bailey and Elkan, 1994), MDscan (Liu *et al*., 2002), a method by Kellis *et al*. (2003), a new conservation-based method by Harbison *et al*. (2004) called CONVERGE, and a modified MEME which was fed conservation information across *sensu stricto Saccharomyces* species. In the main text of this paper we consider only the three programs which do not use conservation information, namely AlignACE, MEME, and MDscan; the supplementary material contains a comparison with all six programs for the TFs considered in this paper, and profiled in all reported environmental conditions. Harbison *et al*. (2004) also do a post-processing step of clustering results from all these programs using cutoffs for significance by various criteria to reach a single motif (if it meets their significance criteria, none otherwise) per TF. Here we compare our results with the raw output from each of the three programs as well as the post-processed single motif derived from all six programs. Thus, our method is competing with six state-of-the-art motif finding algorithms, and also their combination.

There are various differences in the inherent properties of these programs as well as the way in which they are run. AlignACE is based on Gibbs sampling, but uses only single nucleotide frequency to model the background. It was run with the default settings ten times. MEME was run with a fifth order Markov background model using the ZOOPS option and allowed to look for motifs of width 7 to 18 nucleotides. MDscan was also run repeatedly, once with each width in the range 8 to 15 nucleotides.

3.1 Performance of PRIORITY

We set the Dirichlet prior parameters for $\boldsymbol{\phi}$ to 0.5 for all four bases. We gave 3 pseudocounts to γ_k when k is the class of the TF and 1 otherwise. We searched for motifs in the reverse complement of each sequence just as all other programs used for comparison do. With these parameter settings, we applied PRIORITY on each probeset

corresponding to all the 30 TFs profiled under various environmental conditions. Our algorithm was applied for a fixed window size of length 8, so in general it was at a disadvantage with respect to the other programs where the width is varied. We restarted our program 10 times to prevent local optima and report the motif with the highest score.

Table 1 illustrates the results for TFs under the environmental condition considered by Harbison *et al*. (2004) in reporting their final motif. For TFs where they do not report a final motif, we use the probeset resulting from the environmental condition that produces the largest number of bound sequences.

We believe, as is also argued by Liu *et al*. (2002), that a motif finding algorithm should be evaluated based on whether its top motif is correct or not. Each algorithm can use whatever method or score it chooses to rank the motifs and report a top motif. Thus in Table 1, we list the top motif from each of the four algorithms: AlignACE, MEME, MDscan, and PRIORITY according to their respective scoring systems. We also list the final motif reported by Harbison *et al*., but it is important to note that this final motif is produced after considerable human and computational efforts. The post-processing steps include testing multiple motifs from each of the six programs for significance by AUC scores as well as enrichment scores, and then clustering them to produce one motif.

Looking at the table, it is clear that the top motifs from AlignACE rarely match the true motifs from the literature. We believe this happens because AlignACE uses such a simple model to capture features in the background sequence. It has been shown previously that having a higher order Markov model to model the background sequence helps in motif discovery (Liu *et al*., 2001; Thijs *et al*., 2001). The other programs are not disadvantaged by a simple background model as is AlignACE, but in all cases, are outperformed by PRIORITY, as discussed in the remainder of this section.

For more clarity, we categorize the TFs listed in Table 1 into three groups:

- Group I: Literature consensus motif exists, and PRIORITY fails to find such a motif.
- Group II: Literature consensus motif exists and PRIORITY succeeds in finding such a motif.
- Group III: No literature consensus motif exists.

We now discuss TFs falling into these groups in detail.

Group I: This group includes only four TFs: Arr1, Yap3, Yap5, and Yap6. These are all bZip proteins and members of the Yap family (Arr1 is also called Yap8). No program finds motifs matching the literature for any of these four. Thus when PRIORITY fails, the other programs also fail. However, in the case of Arr1, Yap5, and Yap6, PRIORITY predicts a class other than bZip. This is a clue to the fact that the motif the algorithm converges to in these cases may not be a true motif of the TF that was profiled. While we still consider these three cases as failures of our algorithm, at least the algorithm provides some diagnostic information.

Group II: This group includes a total of 20 TFs: Cad1, Cin5, Gcn4, Hac1, Sko1, Yap1, Yap7, Fkh1, Fkh2, Cbf1, Ino2, Ino4, Pho4, Tye7, Leu3, Nrg1, Rap1, Reb1, Ste12, and Ume6. Among the 20 motifs correctly identified by our program, AlignACE finds 2, MEME finds 13, and MDscan finds 17. None of the three other programs finds the true motif for bZip Sko1. While MDscan finds the true motif for Hac1, it does not appear as the post-processed final motif reported by Harbison *et al*.

Along with the correct motif, PRIORITY consistently predicts the true class for TFs in the three classes (100% accuracy). It also correctly assigns the ''other'' class to five of the six TFs not belonging to the three classes explicitly modeled; although PRIORITY learns the true motif of Ste12, it assigns the wrong class. We believe this case is an instance of the algorithm getting stuck in a local maximum or a misclassification by the forkhead binary classifier.

Judging by the performance of PRIORITY on these TFs, we see that despite the computationally expensive steps of Harbison *et al*. in calculating the final motif, our program directly reports better results than the post-processed combination of all six programs.

Group III: Here we consider the remaining six TFs (Cst6, Met28, Met4, Fhl1, Phd1, Sok2) for which there is no known consensus in the literature. For the bZips Cst6 and Met28, without experimental verification, there is no way of knowing for sure if the motifs found by our method are indeed true.

For Met4, Harbison *et al*. find a motif using their algorithm CONVERGE (which exploits cross-species sequence conservation information). This long motif is present in only eight of the 37 bound probes, hence it is no surprise that programs that do not use conservation information are not able to find it. However, we do not know if it is a true motif; in fact, in the literature search that we conducted, we did not find any evidence of Met4 binding DNA directly. Our algorithm finds a different motif for this set of bound intergenic regions which is present in 29 of the 37 sequences and assigns it a bHLH class. This leads us to conclude that this motif could belong to a bHLH protein which is either a cofactor (binds to the same set of sequences separately) or forms a complex with Met4 and binds DNA. Subsequent literature search proves the latter to be true: Met4 forms a complex with Cbf1 and Met28, and it is Cbf1 (a bHLH class protein) which makes contact with DNA at TCACGTG (Kuras *et al*., 1997). PRIORITY does not find the same motif for Met28. In addition to being part of this complex, Met28 is part of other complexes which bind DNA (Blaiseau and Thomas, 1998) and is also capable of binding DNA by itself with low affinity (Kuras *et al*., 1997). We believe these different binding modes dilute the binding site signal.

For forkhead Fhl1, all programs find the same motif (see reverse complement for MEME). This motif is an exact match to the Rap1 binding site. Rap1 does not fall into any of the three classes, and PRIORITY diagnoses this by reporting the class associated with the motif to be ''other'', suggesting that the motif is most likely not a motif for Fhl1. More than half of the probes bound by Rap1 appear in the set bound by Fhl1. Indeed, these TFs are known to be cofactors for some ribosomal protein genes and bind cooperatively (Schawalder *et al*., 2004). We could not find any definitive evidence in the literature either of Fhl1 binding DNA directly, or via a complex with Rap1 or some other TF. However, if Fhl1 does bind DNA directly, and the motif learned is its true motif, one would expect to find multiple copies of the motif (since both Rap1 and Fhl1 need a site on the same probe to which to bind). Harbison *et al*. attempted to determine which TFs tended to use repetitive motifs, but Rap1 does not seem to fall into this category (nor does Fhl1). This makes us believe that the motif learned is bound exclusively by Rap1.

Table 1. Motif comparison for 30 TFs with four different programs. Table shows the motifs learned by various algorithms used by Harbison *et al*. and those learned by our algorithm. For comparison, we use the motifs with the top MAP score for AlignACE, MEME, and MDscan, as well as the final motif reported by Harbison *et al*. after clustering results from these three and three other motif finding programs which use conservation information. In the fifth column we report the top motif according to our score. We also report the predicted class and the percentage of entries in ***c*** contributing to that class. The last column is the literature consensus as used by Harbison *et al*. collected from YPD, SCPD, and TRANSFAC databases at the time their paper was published. The bold sections in the motifs indicate either a match with the literature consensus in the final column or to a motif we found in the literature search we conducted. In cases where the match is not obvious, it is probably because the reverse complement of the sequence matches the literature consensus. Lower case letters in the motifs indicate a weaker preference (less information content at that position). Ambiguity codes: S=C/G, W=A/T, R=A/G, Y=C/T, M=A/C, K=G/T, and '.'= A/C/G/T.

TF	Harbison *et al.*				PRIORITY		Literature
	AlignACE Top MAP	MEME Top MAP	MDscan Top MAP	Post-processed motif from all six programs	Top MAP	Predicted class	
Basic Leucine Zipper TFs							
Arr1	R.AmA.a.A.A.AmA.A	cAmAcACmcAmAmayrcA	CACACACAC	—	YAAACAca	fork 78%	TTACTAA
Cad1	GtGTGTGkGTGTG	GC**TKACTAA**T.	GKGTGTGK	m**TTAsTmA**kC	GC**TTACTA**	bZip 73%	TTACTAA
Cin5	AARAAAA.AA.A	TTTYyttytTy.ytyYYK	.GSGssgG	**TTAygTAA**	**TTAyGTAA**	bZip 94%	TTAC[r]TAA*
Cst6	A.A..rAAA.A..A..a.A	rmAtk.mAwrCRAAaA	AgTY.AsT	—	.ACTGGAC	bZip 80%	—
Gcn4	rAAAAARAAa	yTyTyyTyyYTyTTTc	**TGAsTCAt**	**TGAsTCa**	**ATGACTCA**	bZip 96%	ArTGACTCw
Hac1	A..rAA..MAAARA	TrCSTSkccwywtmM	**TAcGTGkC**	—	**ACGTGTCA**	bZip 76%	kGmCA[G]CGTGTC*
Met28	a.A.A.....AAAA.AAA	TkyTTTTkskssskcTTw	ATrTAyAT	—	SKAAACYG	bZip 75%	—
Met4	AA.AR.RARAAA	AArAAMmmRmAA	TATATATAT	RMmAwsTGKSgyGsc	TGTCACGt	bHLH 80%	—
Sko1	mAAA.RARr.AA	TkTTkyyykTTTkyKKCk	sSgtacSs	—	.t**ACGTCA**	bZip 72%	ACGTCA
Yap1	AwMArrAAr..A	ssTTTyCrT	**TTA.TAA**k	**TTaGTmA**Gc	**TKACTAA**w	bZip 87%	TTAsTAA
Yap3	A.A...A.A.A.A.A.A.Amr	T.kyttcTT.mTTkTT	CACACACAC	—	.CTAAaTS	bZip 65%	TTACTAA
Yap5	GTGTG.GTGTG	GTCGAgsAgAAcsAgGAt	CACACmCAC	—	CGTGKGYG	bHLH 93%	TTACTAA
Yap6	RArAArAAAA	magAAA.rrrAArArR	smYGCAks	—	..CGTGG.	bHLH 91%	TTACTAA
Yap7	AAAARRAAAR	m**TTAsTmA**kc..	**TKAsTMA**k	m**TkAsTmA**k	**TkASTAA**K	bZip 82%	TTACTAA
Forkhead TFs							
Fhl1	RTGTayGGrTG	.t.taCAyCCrTACAYyy	TGyryGGr	rTGTayGGrtg	TGTAYGGR	other 94%	—
Fkh1	RAR.ARA.RAA.A	aaa**.rtAAACAa**..r.a	**tTGTTTAC**	**tTgTTTac**	**GTAAACAA**	fork 95%	GGTAAACAA
Fkh2	rRaAAR.AAA.R	AArrr.rAAAaa.r.AAA	**.GtAAACAA**	aaa**.GTAAACAa**	**GTAAACAA**	fork 94%	GGTAAACAA
Basic Helix Loop Helix TFs							
Cbf1	rAAAAARAAR	**RTCACGTGm**	**kCACGTGm**	**tCACGTG**	**rTCACGTG**	bHLH 97%	rTCACrTGA
Ino2	rARARARR.AA	s**CAYsTGMw**.a	k**CAsrTG**c	**CAcaTG**c	**TCACAT**GC	bHLH 86%	ATTTCACATC
Ino4	A...A.AARrArAR	t**TTYCACATG**s	**CAygTGma**	**CATGTGaaa**a	**CATGTGAA**	bHLH 85%	CATGTGAAAT
Phd1	rRAAARrRAA	rAaA.grAaA.RrRaA	.SSSsSSS	sc.GC.gg	kCGTGsc.	bHLH 95%	—
Pho4	AAAAArAAA..A	**sCACGTg**	**sCACGTGs**	**CACGTGs**	**CACGTG**cs	bHLH 81%	wcacgtk.g
Sok2	ARAArRAAA.R	ArrM..AAAmr.RrAA	SSss.sSG	tGCAg..a	.sCGTG..	bHLH 93%	—
Tye7	rRAARAraAAAYs	rY**CACsTG**AYg	T**CACGTG**A	t**CACGTG**Ay	**CAsGTG**AT	bHLH 92%	CA..TG
TFs belonging to structural classes other than the three modeled explicitly							
Leu3	aA.AAAaAAA...A	**gCCsGtacCGS**wc	**CGgtacCG**	**cCGgtacCGG**	**GgtACyGG**	other 80%	yGCCGGTACCGGyk
Nrg1	rAAAAArAAR	srAarmSrAAA	gGA**CCCT**k	GGa**CCCT**	.GGA**CCCT**	other 89%	CCCT
Rap1	**rTGYayGGrTg**	**..grTGyayGGrTGyr**	**yCCRtrCM**	**tGyayGGrtg**	**ACCCRTAC**	other 90%	wrmACCCATACAyy
Reb1	ksCGGGTAAy	...k**sCGGGTAA**y.	r**TTACCCG**	**CGGGTAA**	m**TTACCCG**	other 93%	TTACCCGG
Ste12	AAAArRAAA..R	gAaACa..t.**TgAaACa**	**tGTTTCA**.	**tgAAAC**	tt**TGAAAC**	fork 95%	ATGAAAC
Ume6	**TsGGCGGCT**A	ww**TAGCCGCCsA**.s	T**AGCCGCC**	t**aGCCGCCs**w	**AGCCGCCs**	other 86%	wGCCGCCGw

* The motif with the inserted r was experimentally confirmed by Harbison *et al*. after they conducted a gel-shift assay to verify the authenticity of the motif they obtained by their *in silico* analysis for Cin5.

** Harbison *et al*. report the longer motif with the central G as literature consensus, but in a literature search we conducted, we found that a new binding site TYACGTGTYM without the central G has been experimentally confirmed by Patil *et al*. (2004) using gel-retardation assays.

For the two bHLH TFs Phd1 and Sok2, the final motifs reported by Harbison *et al*. are both matches to the zinc-coordinating Sut1 TF which does not belong to any of the three classes we studied. Looking at the bound probes, Harbison *et al*. conclude that both pairs Sut1/Phd1 and Sut1/Sok2 are highly co-occurring regulator pairs. This, we believe is a case similar to that of Fhl1, where a strong motif of a different co-occurring TF is learned by regular motif discovery algorithms. The difference is that our algorithm does not find the strong Sut1 motif like it finds Rap1 for Fhl1. Instead, it finds motifs of the bHLH class for both TFs. We thus think these motifs could be true motifs of the two bHLH TFs.

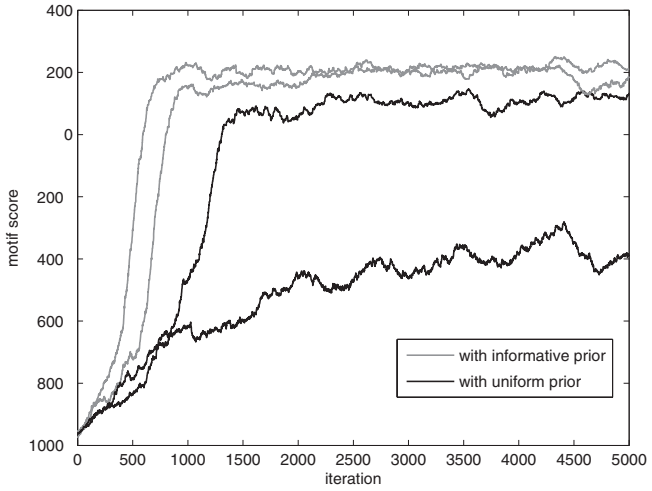


Fig. 2. Motif scores for two Gibbs samplers searching for a Gcn4 motif, one with and the other without the informative prior, over 5000 iterations. Both programs were run five times from different starting locations. The two black plots are the best and worst runs for the program with the uniform prior. The two grey plots are the best and worst runs for the program with the informative prior. Although the absolute values of the scores are not comparable (due to an arbitrary constant value assigned to the uniform prior), it is clear that the number of iterations taken to converge for the algorithm with the informative prior is almost half. Also, each of the five runs converges to a similar final motif in the case of the program incorporating the informative prior. On the other hand, during the worst of the five runs for the program with the uniform prior, the sampler gets stuck in a local maximum that corresponds to a suboptimal motif.

Partitioning the TFs in this manner enables us to draw some important conclusions about the performance of PRIORITY. Simply looking at the results of Group I and Group II, we see that our algorithm finds the correct motif whenever at least one of the other programs finds it and sometimes when none do. From results on TFs in Group III, we see that our program learns motifs of co-occurring TFs and predicts the true class of the co-occurring TF. When the class of the co-occurring TF is different from the profiled TF, our program may help to diagnose the existence of this co-occurring TF.

3.2 Performance of single-class PRIORITY

Sometimes, we know in advance the structural class of the TF which is binding a set of DNA sequences. In such a case, we can fix the class parameter $\boldsymbol{c}$ in advance and not sample from it. We applied this single-class version of PRIORITY on the same ChIP-chip data by setting the class parameter to the respective class of the TF.

Here we do not list the results obtained by using the ''true'' class prior on each of the 30 TFs. The final motifs are not very different, but we notice a big difference in the running times of the sampler when using a single-class informative prior versus using a uniform prior (as is done in most programs). As just one example, we concentrate on Gcn4, a bZip protein, which seems to have a strong motif. Our version of the simple Gibbs sampler with a uniform prior (which is similar to AlignACE with a higher order background model) also finds it.

Figure 2 is a graph of the score of the sampled motif at each iteration (explained in Section 2.4) versus the number of iterations. We ran the sampler with and without the informative prior five times for 5000 iterations and recorded the score of the motif at the end of each iteration. The final motif at the end of each run is simply the motif that scored the best at some point during the run. We have shown the best and the worst scoring runs with and without the informative prior. Although both methods have respective maximum scores at the same values of $\mathbf{Z}$, the sampler with the informative prior converges much sooner than the one with the uniform prior. In fact, in one of the runs, the sampler with the uniform prior gets stuck in a local maximum and remains stuck for all 5000 iterations. With the single-class informative prior, the sampler is less likely to suffer this fate.

4 DISCUSSION

We demonstrate the benefits of using class-specific priors in *de novo* motif discovery problems. More generally, we show how the presence of an informative prior over sequence locations makes it possible to learn the correct motif where conventional methods that use a uniform prior fail.

A novel feature of our method is its ability to output the probable class of the TF binding the motif along with the motif. This gives users more confidence in the learned motif being a description of ''true'' binding sites in cases where the structural class of TF is known. In cases where the TF is not known, the predicted class can be used to limit the possible TFs to be further investigated. For instance, in the case of searching for binding sites in the upstream regions of a set of coexpressed genes, an indication of the class may provide a clue as to which TF could be regulating the set.

In cases where a strong motif of a different TF exists in the same probeset (e.g., Met4, Fhl1), PRIORITY correctly finds this strong motif. In addition, by predicting the class of this motif as the true class which is different from the class of the profiled TF, the program is able to diagnose the presence of the co-occurring TF.

Throughout the paper, we have used PSSMs to model motifs. The PSSM model inherently assumes two things: 1) the binding sites recognized by a particular TF are of fixed length, and 2) position-specific nucleotide preferences exhibit independence between positions. However, experimental and computational studies over the past few years have shown that positions within binding sites are not always independent. Bulyk *et al*. (2002) showed experimentally that for the zinc finger Zif268, there is significant interdependence between the nucleotides of its binding sites. To have a more flexible model for binding sites, Agarwal and Bafna (1998) proposed using Bayesian networks. Since learning general Bayesian networks is an NP-hard problem (Chickering, 1995), Agarwal and Bafna (1998) relaxed their model to trees, and Barash *et al*. (2003) extended this to mixtures of trees and mixtures of PSSMs. Their work showed that these more expressive models indeed yielded better likelihood scores. However, incorporation of a more expressive model into the *de novo* motif finding problem makes the search more complex when no additional information is used. In such cases, when learning a more complex model, an informative prior will prove even more useful in focusing the search significantly.

Our method assigns a prior on the locations within each sequence X_i and not on any specific form of the motif model. Thus in principle, we can incorporate our prior into any general motif finding algorithm and any motif model. Adding a prior on the motif model is orthogonal to our methodology, and can be used when required.

We are the first to propose an informative prior over sequence locations, but others have used structural information to add a prior over motif models (in each case, a PSSM). Sandelin and Wasserman (2004) use JASPAR (Sandelin *et al*., 2004) PSSMs to build a single familial binding profile for each TF family and use that as a prior over PSSMs. However, their work is on narrower domain classes, each not containing more than 10 members. Also, they need to know what family the TF belongs to beforehand. Macisaac *et al*. (2006) extend this concept of DNA-binding profiles to include more families and more variations within families. They generate hypotheses from the profiles and test each one on ChIP-chip data in a classifier-based approach. Xing and Karp (2004) propose a new Bayesian model to capture structural properties typical of particular families of motifs. They learn expressive profiles from PSSMs specific to different classes of TFs. They have results only on simulated data and unfortunately we could not find the code for comparison. Slightly different, but based on the same idea of using prior knowledge related to PSSM models is the SOMBRERO algorithm by Mahony *et al*. (2005). They cluster known PSSMs using self organizing maps (SOMs) and use these clusters as prior knowledge for their search. All these approaches generate a prior over PSSMs and thus apply it on PSSMs directly. Sandelin and Wasserman use pseudocounts to initialize the PSSM they intend to learn, Macisaac *et al*. use their profiles as priors on PSSMs during EM, Xing and Karp use the parameters learned from their profile model as a prior on PSSMs, and Mahony *et al*. use clusters learned from known PSSMs as a starting point for their SOM algorithm which has PSSMs as nodes. Thus these methods can be used only if the motif model to be learned is a matrix based model like a PSSM.

Since we include various features from raw binding sites in our classifiers, we believe we are able to capture inter-position dependencies and structures like palindromes where these other methods cannot. Also, since Sandelin and Wasserman (2004) and Xing and Karp (2004) consider only PSSMs, they lose information about binding sites which were not used to form the PSSM, either because they were of a different size or they just did not contribute to a high scoring PSSM.

Kaplan *et al*. (2005) devise a structure-based approach to predict binding sites from the Cys_2His_2 zinc finger protein family. Their approach is the reverse of ours in the sense that they predict DNA-binding preferences from the zinc finger residue information of the TF and then scan the genome for putative binding sites with those preferences. It is not possible for us to compare our results with theirs due to the difference in the classes under consideration.

Thus far, we have considered only three classes of TFs in yeast. We are in the process of expanding our work to include other big classes like Cys_2His_2, homeodomains, etc. The problem with increasing the number of classes is not only with finding a good binary classifier for each new class, but also the increased computational time required for the Gibbs sampler to converge to sampling from the posterior and visit good optima. For up to two classes, the computational time is fine. In fact, as described in Section 3.2, the sampler reaches its maximum faster with a single-class informative prior than with a uniform prior. However for more than two class-specific priors, we notice the sampler begins to get stuck in local maxima more often. Multiple restarts solves the problem for three classes (the results of which are described in this paper) but it is open at this point how well this will scale to an even larger number of classes. There is a huge body of literature on convergence in Gibbs samplers and other MCMC methods, and we are in the process of exploring other search techniques which may yield faster convergence.

One current disadvantage of our method and all the methods considered by Harbison *et al*. is that none of them provide a significance score to the discovered motif. As a result, the user is left having to calculate various significance scores after the fact based on enrichment, AUC scores, or some other metric as Harbison *et al*. do in their paper. Having multiple priors with different distribution values makes it more tricky. In the case of the single-class version of PRIORITY, a *p*-value can be calculated using random sequence sets of similar length distribution (see supplementary material).

The goal of this study is to demonstrate the significant benefits of informative priors over sequence locations; we have not yet incorporated additional features like learning the optimal width of the motif, searching for multiple copies, etc. We note, however, that these features are useful and will only further improve the performance of the algorithm.

In closing, we believe that using algorithms based only on statistical over-representation will fall short when searching for motifs in more complex organisms having genomes with large intergenic regions. Using informative priors over sequence locations—constructed on the basis of conservation among species (Kellis *et al*., 2003), class-specific DNA binding preferences as presented here, or information like nucleosome occupancy (Lee *et al*., 2004)—will benefit motif finding algorithms as they are applied to more complex organisms.

ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge that the research presented here was supported in part by an Alfred P. Sloan Fellowship to U.O., and by a National Science Foundation CAREER award and an Alfred P. Sloan Fellowship to A.J.H.

REFERENCES

Agarwal,P. and Bafna,V. (1998) Detecting non-adjacent correlations within signals in DNA, *RECOMB '98*

Bailey,T. and Elkan,C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers, *ISMB '94*, AAAI Press, Menlo Park, California, pp. 28–36.

Barash,Y., Elidan,G., Friedman,N., and Kaplan,T. (2003) Modeling dependencies in protein-DNA binding sites, *RECOMB '03*.

Blaiseau,P. and Thomas,D. (1998) Multiple transcriptional activation complexes tether the yeast activator Met4 to DNA, *The EMBO Journal*, 17:6327–6336.

Bulyk,M., Johnson,P., and Church,G. (2002) Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors, *Nucleic Acids Research*, 30:1255–1261.

Chickering,D. (1995) Learning Bayesian networks is NP complete, In Fisher,D. and Lenz,H., eds., *Learning from Data: Artificial Intelligence and Statistics V*, pp. 121–130.

Dempster,A., Laird,N., and Rubin,D. (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39:1–38.

Gelfand,A. and Smith,A. (1990) Sampling based approaches to calculating marginal densities, *Journal of the American Statistical Association*, 85:398–409.

Harbison,C., Gordon,D., Lee,T., Rinaldi,N., Macisaac,K., Danford,T., Hannett,N., Tagne,J., Reynolds,D., Yoo,J., Jennings,E., Zeitlinger,J., Pokholok,D., Kellis,M., Rolfe,P., Takusagawa,K., Lander,E., Gifford,D., Fraenkel,E., Young,R. (2004) Transcriptional regulatory code of a eukaryotic genome, *Nature*, 431:99–104.

Kaplan,T., Friedman,N., and Margalit,H. (2005) Ab initio prediction of transcription factor targets using structural knowledge, *PLoS Computational Biology*, 1(1):e1.

Kellis,M., Patterson,N., Endrizzi,M., Birren,B., and Lander,E. (2003) Sequencing and comparison of yeast species to identify genes and regulatory elements, *Nature*, 432:241–254.

Kim,S., Lund,J., Kiraly,M., Duke,K., Jiang,M., Stuart,J., Eizinger,A., Wylie,B., and Davidson,G. (2001) A gene expression map for *Caenorhabditis elegans*, *Science*, 293:2087–2092.

Krishnapuram,B., Figueiredo,M., Carin,L., and Hartemink,A. (2005) Sparse multinomial logistic regression: Fast algorithms and generalization bounds, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 27:957–968.

Kuras,L., Barbey,R., and Thomas,D. (1997) Assembly of a bZIP-bHLH transcription activation complex: Formation of the yeast Cbf1-Met4-Met28 complex is regulated through Met28 stimulation of Cbf1 DNA binding, *The EMBO Journal*, 16(9):2441–51.

Lee,C., Shibata,Y., Rao,B., Strahl,B., Lieb,J. (2004) Evidence for nucleosome depletion at active regulatory regions genome-wide, *Nature Genetics*, 36(8):900–905.

Liu,J. (1994) The collapsed Gibbs sampler with applications to a gene regulation problem, *Journal of the American Statistical Association*, 89:958–966.

Liu,J., Neuwald,A., and Lawrence,C. (1995) Bayesian models for multiple local sequence alignment and Gibbs sampling strategies, *Journal of the American Statistical Association*, 90:1156–1170.

Liu,X., Brutlag,D., and Liu,J. (2001) BioProspector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes, *Pacific Symposium on Biocomputing '01*, World Scientific, New Jersey, pp. 127–138.

Liu,X., Brutlag,D., and Liu,J. (2002) An algorithm for finding protein-DNA binding sites with applications to chromatin immunoprecipitation microarray experiments, *Nature Biotechnology*, 20:835–839.

Liu,X., Noll,D., Lieb,J., and Clarke,N. (2005) DIP-chip: Rapid and accurate determination of DNA binding specificity, *Genome Research*, 15(3):421–427.

Macisaac,K., Gordon,D., Nekludova,L., Odom,D., Schreiber,J., Gifford,D., Young,R., Fraenkel,E. (2006) A hypothesis-based approach for identifying the binding specificity of regulatory proteins from chromatin immunoprecipitation data, *Bioinformatics*, 22:423-429.

Mahony,S., Golden,A., Smith,T., and Benos,P. (2005) Improved detection of DNA motifs using a self-organized clustering of familial binding profiles, *Bioinformatics*, 21 (Supp 1):i283–i291.

Mukherjee,S., Berger,M., Jona,G., Wang,X., Muzzey,D., Snyder,M., Young,R., and Bulyk,M. (2004) Rapid analysis of the DNA binding specificities of transcription factors with DNA microarrays, *Nature Genetics*, 36(12):1331–1339.

Narlikar,L. and Hartemink,A. (2006) Sequence features of DNA binding sites reveal structural class of associated transcription factor, *Bioinformatics*, 22:157–163.

Patil,C., Li,H., and Walter,P. (2004) Gcn4p and novel upstream activating sequences regulate targets of the unfolded protein response, *PLoS Biology*, 2(8):E246.

Roth,F., Hughes,J., Estep,P., and Church,G. (1998) Finding DNA regulatory motifs within unaligned non-coding sequences clustered by whole-genome mRNA quantitation, *Nature Biotechnology*, 16:939–945.

Sandelin,A., Alkema,W., Engström,P., Wasserman,W., and Lenhard,B. (2004) JASPAR: An open access database for eukaryotic transcription factor binding profiles, *Nucleic Acids Research*, 32(1) Datebase Issue.

Sandelin,A. and Wasserman,W. (2004) Constrained binding site diversity within families of transcription factors enhances pattern discovery bioinformatics, *Journal of Molecular Biology*, 338(2):207–215.

Schawalder,S., Kabani,M., Howald,I., Choudhury,U., Werner,M., and Shore,D. (2004) Growth-regulated recruitment of the essential yeast ribosomal protein gene activator Ifh1, *Nature*, 432:1958–1061.

Siggia,E. (2005) Computational methods for transcriptional regulation, *Current Opinion in Genetics and Development*, 15:214–221.

Spellman,P., Sherlock,G., Zhang,M., Iyer,V., Anders,K., Eisen,M., Brown,P., Botstein,D., and Futcher,B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization, *Molecular Biology of the Cell*, 9:3273–3297.

Staden,R. (1984) Computer methods to locate signals in nucleic acid sequences, *Nucleic Acids Research*, 12:505–519.

Thijs,G., Lescot,M., Marchal,K., Rombauts,S., De Moor,B., Rouze,P., and Moreau,Y. (2002) A higher-order background model improves the detection of potential promoter regulatory elements by Gibbs sampling, *Bioinformatics*, 17:1113–1122.

Thijs,G., Marchal,K., Lescot,M., Rombauts,S., De Moor,B., Rouze,P., and Moreau,Y. (2002) A Gibbs sampling method to detect over-represented motifs in the upstream regions of coexpressed genes, *Journal of Computational Biology*, 9:447–464.

Wasserman,W. and Sandelin,A. (2004) Applied bioinformatics for the identification of regulatory elements, *Nature Reviews Genetics*, 5(4):276–287.

Wingender,E., Chen,X., Fricke,E., Geffers,R., Hehl,R., Liebich,I., Krull,M., Matys,V., Michael,H., Ohnhauser,R., Pruss,M., Schacherer,F., Thiele,S., Urbach,S. (2001) The TRANSFAC system on gene expression regulation, *Nucleic Acids Research*, 29:281–283.

Xing,E. and Karp,R. (2004) MotifPrototyper: A Bayesian profile model for motif families, *Proc. Natl. Acad. Sci.*,101:10523–10528.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e393–e401
doi:10.1093/bioinformatics/btl245

Apples to apples: improving the performance of motif finders and their significance analysis in the Twilight Zone

Patrick Ng[1], Niranjan Nagarajan[1], Neil Jones[2] and Uri Keich[1,*]

[1]Department of Computer Science, Cornell University, Ithaca, NY, USA and [2]Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA, USA

ABSTRACT

Motivation: Effective algorithms for finding relatively weak motifs are an important practical necessity while scanning long DNA sequences for regulatory elements. The success of such an algorithm hinges on the ability of its scoring function combined with a significance analysis test to discern real motifs from random noise.

Results: In the first half of the paper we show that the paradigm of relying on entropy scores and their E-values can lead to undesirable results when searching for weak motifs and we offer alternate approaches to analyzing the significance of motifs. In the second half of the paper we reintroduce a scoring function and present a motif-finder that optimizes it that are more effective in finding relatively weak motifs than other tools.

Availability: The GibbsILR motif finder is available at http://www.cs.cornell.edu/∼keich

Contact: Uri Keich, keich@cs.cornell.edu

1 INTRODUCTION

The identification of transcription factor binding sites, and of *cis*-regulatory elements in general, is an important step in understanding the regulation of gene expression. To address this need, many motif-finding tools have been described that can find short sequence motifs given only an input set of sequences. The motifs returned by these tools are evaluated and ranked according to some measure of statistical over-representation, the most popular of which is based on the information content or entropy [17] (see [19] for a recent comparative review).

Keich and Pevzner [10] define a *twilight zone* search as one in which there is a non-negligible probability that a maximally scoring random motif would have a higher score than motifs that overlap the ''real'' motif (in the model considered there, a ''real'' motif is implanted into randomly generated background sequences). In such cases, even if one had access to a hypothetically ideal finder that was guaranteed to return the highest scoring alignment in the dataset, the motif might remain unfound. Locating the twilight zone is necessary in deciding whether or not the current state of the art in motif finding is good enough: if existing tools find the correct motif for datasets all the way into the twilight zone in a reasonable time, further improvement will yield at best marginal returns. Of course, improving motif finding tools to be effective into the twilight zone is not merely a theoretical exercise: a biologist searching for regulatory motifs in DNA sequences would generally prefer to choose longer rather than shorter regions in order to avoid missing regulatory elements that are far away from the transcription start site of a gene. The longer the input sequences are, the more likely they are to contain high scoring random motifs, pushing the biologically valid motifs into the twilight zone.

Most existing motif finders can be divided into two classes depending on how they model a motif. Tools that rely on a combinatorial model of a motif define a motif to be a consensus sequence with an associated distance (usually Hamming distance), as described in [14]. Under this definition, the problem of finding motifs in random sequences is mostly solved. The statistics of optimal random motifs are well understood in this context, which led to the characterization of the twilight zone [10]. Moreover, the PatternBranching tool [16] exhibits good performance, even in the twilight zone for reasonable choices of parameters leaving little motivation for further improvement.

On the other hand are tools that describe a motif as a profile, a probabilistic distribution generally modeled with a position weight matrix (PWM). Prominent examples of this class are MEME [1], CONSENSUS [6] and the various approaches to Gibbs sampling (e.g. [11],[13],[7]). Under this definition of a motif, there has been no definitive demonstration of any particular tool's dominance. Moreover there is no reliable characterization of the distribution of optimal random motifs[1], nor is the twilight zone completely understood.

In most applications of a motif finder, the user must decide whether or not a motif reported from a motif finder warrants further biological investigation based on its statistical significance. The first half of this paper deals with the significance analysis of the ubiquitous entropy score. We begin by showing that the common practice of using the *E*-value of the entropy score (defined below) to evaluate the significance of an alignment reported by a motif finder can lead to undesirable results in twilight zone searches. We then discuss two additional intuitively motivated measurements of statistical significance and some pitfalls in their application to motif finding. The second half of the paper discusses an alternative scoring scheme. This is motivated by the observation that comparing entropy scores across different motif finders often leads to inconsistent results regarding the identification of the implanted motif. We reintroduce the Incomplete (data) Likelihood Ratio (ILR) and show it is a better classifier

*To whom correspondence should be addressed.

[1]Some progress was made recently by Frith, *et al*. [5], but the analysis presented there only holds for a small number of sequences.

when it comes to predicting overlap with implanted motifs. This motivates GibbsILR, a new variant of the Gibbs sampler that attempts to maximize the ILR rather than the entropy score.

2 ARE MOTIF FINDERS PSYCHIC? THE CONUNDRUM OF *E*-VALUES

One of the key measurements in determining if a motif finder has identified an important motif is the *E*-value of the entropy score defined as follows. The entropy score or information content of the reported alignment is defined as [17]:[2]

$$I := \sum_{i=1}^{w} \sum_{j=1}^{A} n_{ij} \log \frac{n_{ij}/n}{b_j},$$

where w is the motif width, n_{ij} denotes the number of occurrences of the jth letter in the ith column of the alignment, b_j is the background frequency of the jth letter[3], n is the number of sequences in the alignment, and A the alphabet size ($A = 4$ in this paper). Introduced originally in this context as the ''expected frequency'' [6], the *E*-value is the expected number of random alignments of the same dimension that would exhibit an entropy score that is at least as high as the score of the given alignment. When the *E*-value is high, one can have little confidence in the motif prediction, and conversely when the *E*-value is low, one can have more confidence in the prediction. It is computed by multiplying the number of possible alignments by the *p*-value of the alignment. The latter is defined as the probability that a single *given* random alignment would have an entropy score $\geq$ the observed alignment score. Assuming the customary iid (independent identically distributed) random model the *p*-value can be computed accurately using techniques we previously described [12].

To assess the performance of motif finders in twilight zone searches, we designed the following experiments containing 400 data sets (see the COMBO experiments in the Methods section). Each randomly generated data set contained a deliberately implanted profile motif in such a way that for a non-trivial percentage of datasets, the motif finders we considered would pick motifs that would not overlap the implants. Thus, it is not surprising that the *E*-value of the implanted motif is relatively high. However, with a median *E*-value of 8×10^{15} it seems this problem is way beyond the twilight zone. Indeed, one would suspect that in this case even the ideal finder would not be able to pick out an alignment with significant overlap to the implanted motif from the large number of background alignments with better entropy score. Rather startlingly, exactly the opposite is true: of 400 data sets, the Gibbs sampler [11] found an alignment overlapping more than 30% of the implanted sites in 288 cases[4]. It is important to note that these data sets are constructed exactly according to the model used in computing the *E*-values, thus we can safely assume the *E*-value is quite accurate [12].

How can our motif finders be so lucky that they pick a ''real'' motif out of such a huge haystack? A partial answer to this riddle is obtained by noting that when a motif is implanted into a set of long sequences, there is a good chance that a random string in one of the sequences will slightly improve the entropy score. Of the 288 data sets for which the Gibbs sampler found an overlapping alignment (above the 30% threshold), the median *E*-value of the reported motif was 8.7×10^{11} or 4 orders of magnitude better than the initial motif. Still, it is a very impressive haystack and a more complete answer probably lies in what we do not see: how many alignments that overlap with our implant have a score as good as the one found? These high scoring ''satellite'' alignments define some ''domain of attraction'' for a motif that is difficult to characterize analytically. Presumably, its size has to be of the order of the *E*-values as sampling optimization procedures such as Gibbs somehow find it. We remark that characterizing this domain of attraction is a potential way to describe the twilight zone of a profile-based motif.

Whatever the explanation is, it is clear that the *E*-value offers little benefit in analyzing the significance of twilight zone search. We next explore alternative approaches to this problem.

3 ALTERNATIVE SIGNIFICANCE ANALYSES

One alternate measure of significance suggested by Hertz and Stormo [6] is that of the ''overall *p*-value''—or $OPV(s)$—of an entropy score s. It is defined as the probability that a random sample of the same size as the input set will contain an alignment of the same dimensions that scores at least as high as s. While this statistic is intuitively appealing, its use faces two hurdles. On the one hand, at present it is all but impossible to calculate $OPV(s)$ for moderately large datasets: even generating an empirical estimate of the OPV would necessarily require the ability to reliably find the highest scoring alignment in any given sample, which cannot be guaranteed for realistic problem sizes. On the other hand, even if an accurate method for calculating $OPV(s)$ were known, the evidence presented next suggests that this significance measure would impose too high a barrier on the entropy score for functional motifs to be distinguishable from noise.

The value of $OPV(s)$ may be conservatively estimated by the probability that at least one of several motif finders would find an alignment of score $\geq s$ in the random data. The point is the latter is amenable to Monte Carlo estimation. Using 1600 randomly generated datasets with no motif implanted we obtain an empirical estimate of the 0.95 quantile of the latter distribution; this is the minimal value s_0 such that for 95% of the datasets all our finders report a top alignment of score $\leq s_0$. We then use s_0 as an empirically derived conservative estimate of the threshold s_1 such that $OPV(s_1) = 0.95$. That is, 95% of the top scoring noise alignments have entropy less than or equal to s_1 and $s_1 \leq s_0$ with high probability. When this derived 5% significance level was applied as a threshold for significance of the 400 data sets in the COMBO experiment, nearly 90% of the correct runs of the Gibbs sampler (i.e., those runs that overlapped the implanted motif by more than 30%) were classified as noise. Since s_0 the conservatively estimated 0.95 quantile is very likely to be greater than the true quantile s_1, this should become more pronounced with better approximations of $OPV(s)$ suggesting it is also too conservative.

One can see that $1 - OPV(s)$ is the distribution function of the ideal motif finder. This raises the natural extension of using a finder-specific OPV: $1 - F_f(s)$ where F_f is the null distribution of the score of the optimal alignment detected by the particular finder. That is, we ask for the probability that the finder will find an

[2]Strictly speaking, relative entropy is defined as I/n.

[3]Typically estimated from the entire sample.

[4]See the Methods section for the parameter setting.

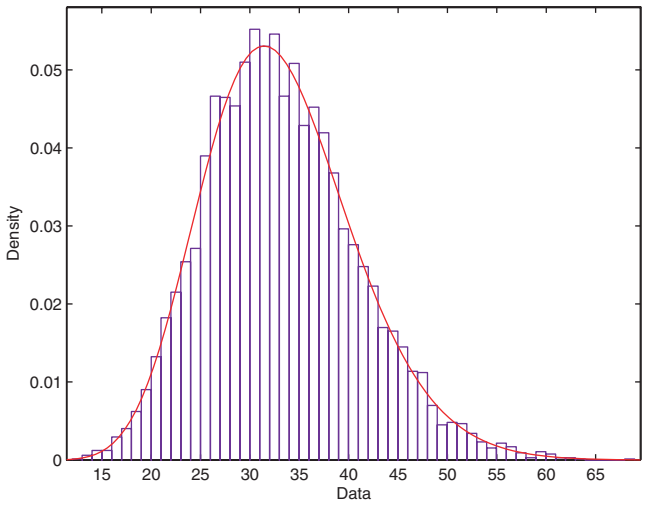


Fig. 1. Shifted gamma fit to 6400 runs of Gibbs with parameters `13 -t100 -L100` on 40 random sequences of length 750, uniformly distributed with no implanted motif.

alignment scoring $\geq s$ in a random dataset (of the same dimensions). Again, we can estimate the quantiles of this distribution function through a Monte Carlo generated empirical distribution. In this case we found that the .95 quantile threshold of Gibbs, estimated from 1600 datasets, yields 13 false positives (FP) and 228 true positives (TP) when applied to the same 400 data sets of the COMBO experiment.[5]

While the empirical distribution can be extremely useful in analyzing the significance of a motif finder's output, generating it *a priori* is typically impossible due to the large number of combinations of parameters. Similarly, generating even a rough estimate of a 0.95 quantile per problem instance is impractical as it would require at least 100 additional runs of the motif finder on a dataset of the same size as the input.

However, if we can characterize the distribution as belonging to some parametric family of distributions, we might do better to estimate the parameters of the distribution rather than directly estimating the quantiles of the distribution. The (limiting) distribution of a maximal ungapped pairwise alignment between two sequences is a Gumbel Extreme Value Distribution (EVD) [9]; the same distribution is encountered empirically in the gapped case and it is presumed to underly the distribution of scores when local multiple alignments are scored according to a presumed phylogeny [15] and in Frith *et al*. [5], which specifically discusses motif finding. Oddly, the empirical null distribution of the reported entropy score for several motif finders exhibited a better fit to a (shifted) Gamma distribution than to the intuitively more appealing Gumbel distribution (see Fig. 1 and Fig. 2 for an example involving Gibbs).[6]

[5]That we expect $5\% \cdot 400 = 20$ false positives and see only 13 is reasonable since some of those random datasets containing high-scoring alignments are masked by higher-scoring motifs that overlap the implant.

[6]To fit a shifted gamma distribution for each shift we find the likelihood of the shifted data by applying a standard maximum likelihood gamma fit to it, and then use a simple one dimensional search of the shift that yields the highest likelihood.

Naturally, the parameters of the distribution could depend on the size and background distribution of the dataset, as well as the parameters used in the motif search. It is surprisingly easy to get a good approximation of how the empirical distribution of CONSENSUS behaves with respect to these variables. The key is to consider the distribution of the *E*-value of the best reported entropy rather than the entropy itself. This *E*-value is fairly stable for a wide range of sequence lengths (375–1500), motif widths (10–50), and with different sequence composition (uniform background *versus* a biased background of (0.2, 0.2, 0.3, 0.3)), see Figure 3 and Table 1. That is, the empirical distribution depends primarily on N, the number of sequences; and q, the number of sub-alignments that CONSENSUS keeps at each stage. While the parameter estimates for the shifted gamma distribution are not perfectly stable, they could be readily improved by dividing the range of parameters (e.g., motif width) into several segments and using one shifted gamma distribution per interval.

We demonstrated above that the OPV or equivalently the distribution function of the ideal finder seems too conservative for estimating the significance of a motif finder's output. Nonetheless it is useful in delineating the twilight zone, which in turn is important for understanding to what extent existing tools might be theoretically improved upon. Indeed, by comparing the empirical distribution of a motif finder with that of the ideal one for a given set of parameters, we can assess the efficiency of the finder for these parameters. It is thus interesting to determine whether this distribution can be approximated by a parametric family. As above we find the surprising result that a shifted gamma distribution gives a better fit to the empirical distribution than a Gumbel distribution. One might expect that the result of maximizing over all possible alignments would naturally result in an EVD but according to our observations this is not the case (see Figure 4). One reason is that the high scoring alignments are heavily dependent, an observation made by Frith *et al*. [5] when trying to explain the less-than-perfect fit they got to a Gumbel distribution.

4 INCOMPLETE (DATA) LIKELIHOOD RATIO AND GIBBSILR

A good scoring function should separate as much as possible real motifs or, in the context of our model, alignments that have overlap with the implant, from purely random ones. The entropy score is the one chosen by popular motif finders such as MEME [1], CONSENSUS [6] and Gibbs Sampler [11]. The latter two specifically try to optimize this scoring function, while MEME uses it only to rank and analyze the significance of its output. It is thus tempting to assume that if we run, for example, both CONSENSUS and Gibbs and take the higher scoring motif we would do better than if we ran each one of them separately. Amazingly, this might not be the case, especially in twilight zone searches. In particular, in our COMBO experiment we find that in 380 of the 400 datasets CONSENSUS finds a motif with higher entropy score than Gibbs, yet Gibbs reports more motifs that have $\geq$30% overlap with the true implant (290 of the sets for Gibbs compared to 208 for CONSENSUS). Comparing the entropy score from different motif finders is thus not an apples to apples comparison as one would expect—somehow it matters how the entropy is maximized. This led us to ask if other scoring functions would possibly capture better the nature of real (implanted) twilight zone motifs. One

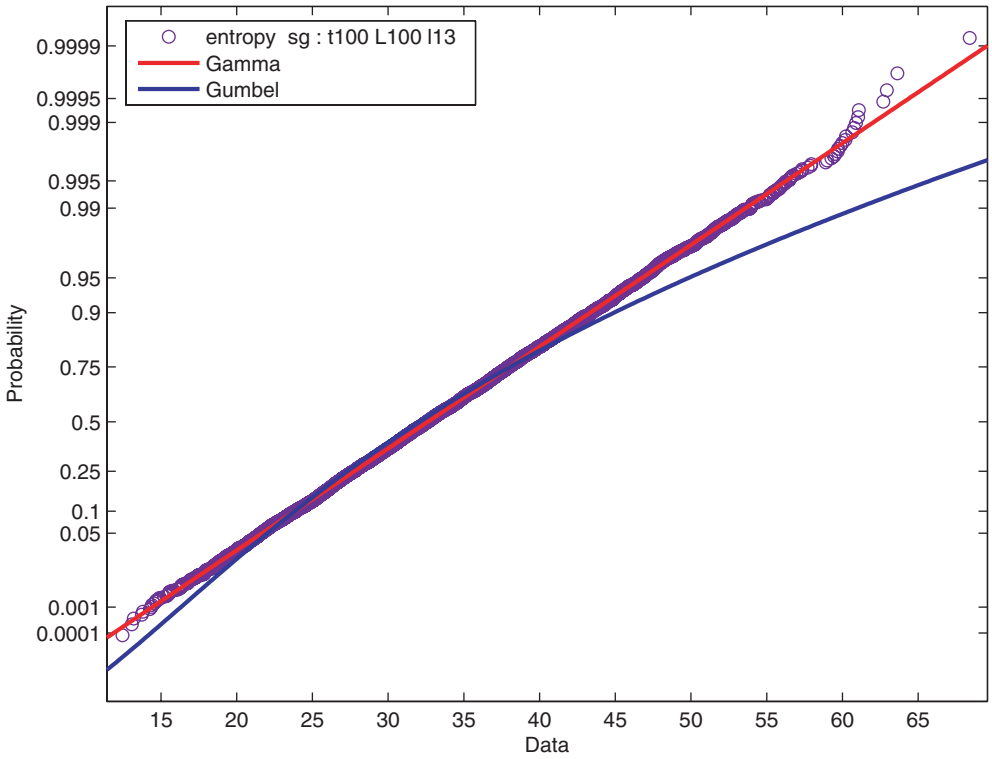


Fig. 2. The probability plot of the fit of a shifted gamma distribution and of a Gumbel Extreme Value Distribution to the data collected in Fig. 1.

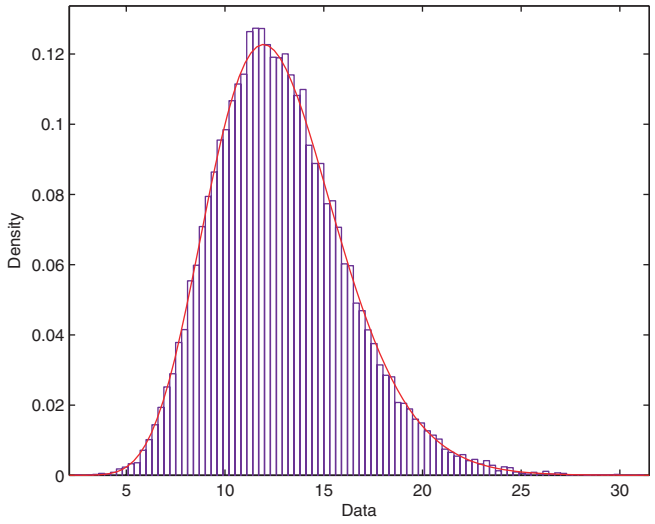


Fig. 3. A (shifted) gamma distribution fitted to the empirical null distribution of $-\log E$-value of the best motif. The data was compiled from 6400 datasets of 40 uniformly distributed sequences of length 750. For each $w \in \{10, 13, 16, 20, 30, 50\}$ CONSENSUS was ran on each dataset with $q = 1000$.

scoring function presented next shows consistent, and at times considerable improvement over the entropy score.

Given a set of N sequences $S = \{S^1, S^2, \ldots, S^N\}$, the formal definition of the Motif Finding problem is to find the set of starting positions within each sequence that corresponds to the location of an implanted motif. We assume there is a profile matrix $\Theta = (\Theta_{ij})$ of length w that represents the implanted motif model, and another profile matrix $\Theta_0 = (\Theta_{0j})$ that represents the background. We define the Incomplete (data) Likelihood Ratio as follows:

$$ILR(\Theta) = \prod_{n=1}^{N} \left[\sum_{m=1}^{|S^n|-w+1} \frac{P(S^n_{m:m+w-1}|\Theta)}{P(S^n_{m:m+w-1}|\Theta_0)} \cdot \frac{1}{|S^n| - w + 1} \right] \quad (1)$$

Intuitively, $ILR(\Theta)$ is the likelihood ratio between two competing hypotheses. The null hypothesis is that the data was entirely generated under the null model Θ_0. The alternative hypothesis is that the data was generated under the OOPS (one occurrence per sequence) model [2] using the motif Θ and the background Θ_0. Unlike the standard entropy score, the ILR scores a motif by taking into account all of the data in S, rather than only the data within a particular alignment. The EM algorithm optimizes the ILR [3], and by extension MEME does as well. However, MEME ranks motifs by entropy and assesses the reported motif's significance through the E-value of the entropy score. In particular, the ILR score has not been previously used to score and rank motifs.

Our tests (described in the Results section) demonstrate that for twilight zone searches ILR is a consistently better classifier than the entropy score for identifying motifs that overlap the implant. Since this holds for all the finders that we tested, most of which optimize the entropy score, it motivates the design of a new finder that tries to optimize the ILR: GibbsILR is based on the Gibbs-sampling technique described by Lawrence *et al.* [11]. Here we modify the original Gibbs sampling strategy by using a hybrid optimization procedure. The Gibbs-sampling motif finder begins each run by picking a random starting position in each sequence in the data

Table 1. Comparison of empirical distributions with the fitted gamma distribution in Figure 3. A column with heading $r(x)$ contains the ratio of the CDF of the empirical distribution to that of the fitted shifted gamma distribution at the x^{th} quantile of the fitted distribution. Note that when considering E-values, small quantiles are good

Test Set	w	r(0.1)	r(0.05)	r(0.01)
40 sequences of length 750	10	0.91	0.83	0.89
to which the gamma was fitted	13	1.13	1.21	1.22
	16	1.06	1.11	1.19
	20	1.10	1.19	1.44
	30	1.04	1.12	1.37
	50	0.65	0.67	0.87
40 sequences of length 1500	10	0.34	0.30	0.44
	13	0.73	0.67	0.94
	16	0.98	1.10	1.37
	20	0.94	0.91	1.19
	30	0.85	0.96	1.25
	50	0.51	0.59	0.75
biased composition	10	0.55	0.56	1.00
20 sequences of length 375	13	0.82	0.82	0.75
and 20 sequences of length 750	16	0.96	0.95	1.19
	20	1.13	1.24	2.12
	30	0.88	0.94	1.44
	50	0.49	0.51	0.37

set. The algorithm then iterates between two steps, commonly referred to as the predictive update step and the sampling step. In the k-th iteration, the predictive update step computes a motif model Θ^k based on the current chosen set of starting positions.[7] The sampling step in turn randomly selects new candidate starting positions with probability proportional to the likelihood ratio of the position given the current model Θ^k.

A well-known property of Gibbs-sampling algorithms is that they are guaranteed to sample the global maximum given sufficient time, but this may take an unacceptably long time to happen. Instead, when the objective function is apparently not making any headway, we can ''restart'' the sampling procedure by initializing a new, independent, Gibbs-sampling run using a new set of random starting positions. Unlike previous Gibbs-sampling motif finders, GibbsILR runs an EM (Expectation-Maximization) algorithm that locally optimizes ILR on the final motif of each Gibbs-sampling run. GibbsILR then produces a motif that exhibits locally optimized ILR score by taking the highest ILR-scoring motif among all of the final motifs derived from the EM step. Finally, for each sequence in the dataset S, the motif instance corresponds to the position with the highest likelihood ratio with respect to the highest ILR-scoring motif profile.

5 RESULTS

The first group of results is based on extensive tests of the performance of six profile-based motif finders on synthetic data. Each of these randomly generated datasets contained a deliberately implanted profile motif (see the Methods section for more details). The output of each of the finders we considered (CONSENSUS, Gibbs, GibbsILR, GLAM, ProfileBranching, and MEME) was post-processed to yield both the entropy and ILR scores of the finder's top reported alignment. We then asked which of these two scores is a better predictor of overlap with the implant (which is a surrogate for a real motif).

We compare the entropy and ILR score by measuring the area under the ROC curve [18], or discrimination, for each finder under the two scoring functions. We classify a set of motif sites as negative if the overlap score is below 0.1; otherwise, we classify it as positive. Intuitively, given a random pair of positive and negative set of profile sites, the aROC tells us the probability of the test correctly identifying the pair's classification. The tests (Table 2) using ILR score have consistently better discrimination than the tests using entropy score. The reader should note that it is however unfair to compare the performance of the finders using aROC, because the number of negatives and positives differ across the finders. For example, GibbsILR has lower discrimination than MEME for both entropy and ILR in COMBO, but GibbsILR has 324 positives to discriminate whereas MEME has only 70 positives.

Similarly we can ask how many true positives (TP) are in the test set if we are willing to accept exactly 10 false positives (FP). Table 2 shows that ILR consistently has higher counts of such TPs than entropy. Moreover, if we would like to design a classifier that only accepts 10 FPs, this analysis shows that the combination of ILR score and GibbsILR would give us the highest number of TPs.

We next combine five motif finders: CONSENSUS, Gibbs_ss, GLAM, MEME, and ProfileBranching by choosing the set of motif sites from the finder with highest ILR. Likewise, we employ the same technique with the entropy. We found that the ILR variant of the combined-finder can perform better than any of its individual finders alone. In the COMBO experiment, the ILR variant found the implants in 311 datasets (i.e. overlap score greater than 0.1), whereas its best individual finder, which is Gibbs_ss, found the implants in only 302 datasets. In the same experiment, the entropy variant found the implants in 291 datasets, which is worse than its best individual finder. For a different approach to combining the output of multiple motif finding algorithms, see [8].

As an additional source of evidence for the utility of the ILR score we generated synthetic data sets implanted with motifs that were verifiably in the (entropy score) twilight zone. The branch and bound algorithm described in the Methods section was then used to find the motif with the optimal entropy score and the ILR score of that motif. Then, based on the results from 1000 such runs we asked the following question: which of the scores, entropy or ILR is a better predictor of overlap with the implanted motif? For the twilight zone data sets that we tested, ILR is consistently better than the entropy score as a predictor of overlap (as measured by the aROC score, with overlap being defined as an overlap score greater than 0.1). As a specific example, for $N = 14$, $L = 80$ and `SHORT` (see Table 3), the entropy score has an aROC score of 0.52 as compared to 0.60 for the ILR score. In practical terms, for a threshold that allows 50 false positives, the ILR score gives 143 true positives as opposed to 101 for the entropy score. Interestingly, in this example, while the ILR score has a positive

[7]The model Θ is inferred from the starting positions by the rule $\Theta_{ij} = \frac{c_{ij} + b_i}{N - 1 + \sum_j b_j}$ where c_{ij} is the count of letter j in the i-th sequence of the alignment and b_j is an *a priori* chosen pseudocount to avoid 0 probabilities.

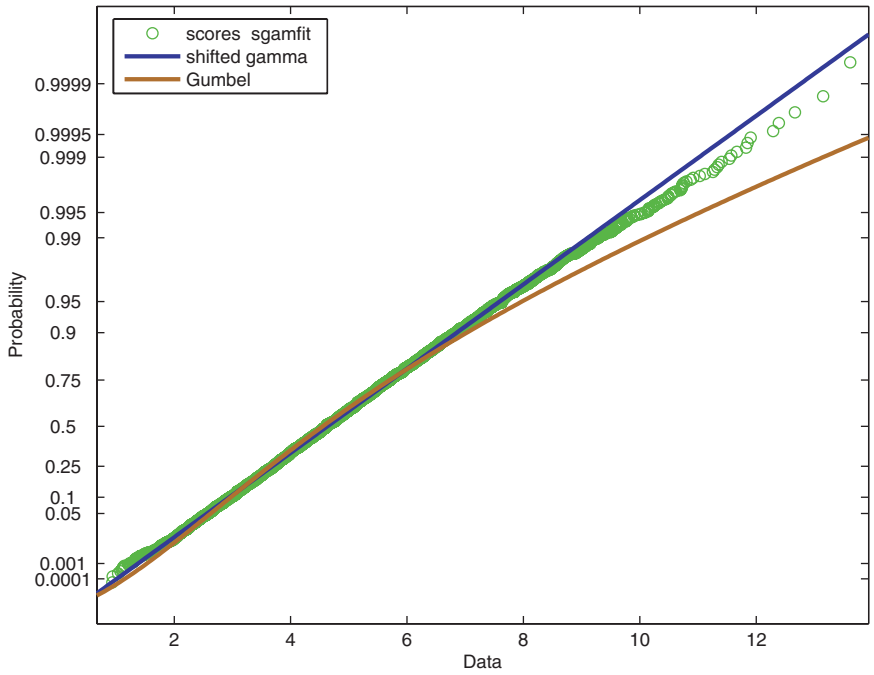


Fig. 4. The probability plot of a fit of the OPV distribution to a shifted gamma distribution and to an EVD distribution; the OPV distribution was generated from the output of the ideal motif finder (searching for motifs of width 7; see Methods section) run on 10000 datasets composed of 10 uniformly distributed sequences with length 100.

Table 2. aROC and only accepting 10 FPs. The column >10% contains the number of datasets that score above the 0.1 overlap threshold. The column TPs contains the number of true-positives in a test if it is willing to accept ≤10 FPs

Experiment	Finders		entropy		ILR	
		>10%	aROC	TPs	aROC	TPs
COMBO	CONSENSUS	223	0.88	154	0.93	169
	Gibbs_ss	302	0.88	208	0.91	231
	GibbsILR	324	0.85	254	0.90	258
	GLAM	170	0.90	117	0.94	127
	MEME	70	0.90	43	0.92	48
	ProfileBranching	222	0.95	183	0.96	190
FIFTY	CONSENSUS	27	0.73	5	0.85	13
	Gibbs_ss	87	0.94	70	0.96	76
	GibbsILR	186	0.96	171	0.96	171
	GLAM	116	0.94	91	0.89	84
	MEME	4	0.64	0	0.73	0
	ProfileBranching	8	0.60	1	0.72	1

Spearman correlation, the entropy score has a statistically significant negative correlation with the overlap score (Spearman correlation p-value of $5.2 \cdot 10^{-4}$)[8].

Finally while not an objective demonstration of the advantage of ILR, GibbsILR did show improvement in our experiments over the other five finders we tested. Fig. 5 shows the overlap distribution for the various finders. For example, the bars at 0.1 are the number of datasets that a particular motif finder found with overlap score between 0.1 and 0.2. GibbsILR finds the most datasets above 0.1 overlap score for both experiments. In the case of FIFTY (Fig. 5b), GibbsILR is significantly better. Note that we tried to equalize the running time of all the algorithms in the benchmark as described in the Methods section below.

[8]Recall that the detected motif was optimized for the entropy, rather than the ILR.

6 CONCLUSION AND FUTURE WORK

We have demonstrated several discouraging observations regarding the use of E-values to determine the statistical significance of a motif. However, the E-value of the entropy score has at least two redeeming qualities. First, it is always conservative, so a motif exhibiting a very low E-value is probably significant. Second, the distribution of the overall p-value for CONSENSUS was easy to characterize when considering the E-value of the entropy score, rather than the entropy score itself. We acknowledge the possibility that other motif finders may exist (that we have not yet tried) whose finder-specific OPV may be similarly quantified.

We have also presented an alternative scoring function to be used in place of entropy, along with a motif finder that uses this function to achieve demonstrably better results than existing algorithms. The motif finder described here is admittedly simplistic, so it seems likely that more effective algorithms could be developed.

We were surprised to discover that the empirical distribution of optimal alignment scores displayed by each algorithm, including the exhaustive motif finder, fit a gamma distribution far better than the intuitively more appealing EVD. It would be informative to determine why the gamma distribution, specifically, seems to model this problem accurately.

Table 3. The position weight matrices used in these experiments

	COMBO				FIFTY				SHORT			
Pos.	A	C	G	T	A	C	G	T	A	C	G	T
1	0.95	0.00	0.00	0.05	0.50	0.00	0.00	0.50	0.95	0.00	0.05	0.00
2	0.00	0.50	0.50	0.00	0.00	0.50	0.50	0.00	0.00	0.05	0.95	0.00
3	0.70	0.10	0.10	0.10	0.50	0.50	0.00	0.00	0.29	0.29	0.21	0.21
4	0.00	0.70	0.30	0.00	0.50	0.00	0.50	0.00	0.00	0.00	0.50	0.50
5	0.50	0.00	0.00	0.50	0.50	0.50	0.00	0.00	0.00	0.00	0.05	0.95
6	0.25	0.25	0.25	0.25	0.00	0.50	0.50	0.00				
7	0.95	0.00	0.00	0.05	0.00	0.50	0.00	0.50				
8	0.25	0.25	0.25	0.25	0.00	0.50	0.00	0.50				
9	0.70	0.10	0.10	0.10	0.50	0.00	0.50	0.00				
10	0.00	0.50	0.00	0.50	0.00	0.50	0.50	0.00				
11	0.00	0.70	0.00	0.30	0.50	0.50	0.00	0.00				
12	0.70	0.10	0.10	0.10	0.00	0.50	0.50	0.00				
13	0.00	0.50	0.50	0.00	0.00	0.50	0.00	0.50				

Finally, an extensive study of the distributional properties of the ILR is a necessary condition for it to become a widely adopted scoring scheme.

7 METHODS

To test the efficacy of any given motif finding algorithm, N independent sequences of length m were sampled by choosing symbols at random from the four letter DNA alphabet corresponding to an iid model for the background frequency. A position was chosen uniformly at random from each sequence and an instance of a profile Θ, generated as described below, was inserted in that position. Thus, the total length of each sequence is $L = m + w$ where w is the length of the motif. A profile is represented as a position weight matrix, a $4 \times w$ array of numbers where Θ_{ij} denotes the frequency of letter i in column j in all aligned instances of Θ. Since we wanted to have control over the implanted motifs the instance were essentially generated by permuting the columns of the alignment. Each column of the alignment matched the corresponding column of the profile up to discretizing effects.

The parameters N and L were chosen such that the motif finders we considered would have a non-trivial percentage of failures (i.e. datasets where they pick motifs with no overlap with the implants). As we allowed our finders to run for a fairly generous amount of time there is reason to suspect that at least some of those failures can be attributed to twilight zone searches [10], in which random alignments with no overlap with the implants score as high as the best motif that overlaps the implant.

Two of the experiments that we report here were generated according to the following rules:

(1) `COMBO`: The motif in this experiment has length 13 with two degenerate columns (6 and 8) as seen in Table 3. Each dataset has 40 sequences of length 1485 + 13.

(2) `FIFTY`: Each column in the motif consists only of two equally probable nucleotides. Each dataset has 40 sequences of length 1485 + 13.

In each experiment, 400 datasets were generated for a given profile, and various motif finding algorithms were run with parameter settings that allowed each motif finder to take from 8–10 minutes to place all motif finders on an equal footing. However, the MEME motif finder does not employ any parameters that allow the control of running time (MEME generally runs in much less than 8 minutes on each data set), so the generally poor performance of MEME compared to the other motif finders is not a reflection of MEME employing a bad algorithm but a reflection of a design decision to place a strict limit on the total amount of time MEME takes. The motif finders used in this study consisted of MEME [1] (`-mod oops -nmotifs 1 -w 13 -dna -text -maxsize 1000000`), the Gibbs Sampler run in Site Sampler (''Gibbs_ss'') of [11] (`13 -d -n -t280 -L200`), Gibbs altered to use the ILR scoring function (''GibbsILR'', `13 -t 250 -L 200 -p 0.05`), GLAM [5] (`-n50000 -r10 -1 -z -a13 -b13`), CONSENSUS [6] (`-L 13 -c0 -q 3000`), and ProfileBranching [16] (`-l 13 -verbose`). We note that Gibbs_ss is our version of the original algorithm optimized for site sampling mode, resulting in a three-fold improvement in running time. For this reason, the results of Gibbs_ss are better than the results of the original algorithm for a fixed running time. All experiments were run under Red Hat Enterprise Linux 4 on a cluster with nodes that have AMD 248 2Ghz 64-bit processors with 2GB RAM and 1GB swap.

The p-value of the entropy of the highest-scoring reported motif was computed by the sFFT algorithm described in [12]. An estimate of overlap for each data set and for each motif finder was computed in the following manner: Let a_n be the position of the implanted motif instance in S_n, and let $\hat{a_n}$ be the position of the motif reported by a motif finder. We define the *overlap* of a motif finder's prediction as:

$$\max_{|i| \le \frac{w}{2}} \left\{ \frac{w - |i|}{w} \cdot \frac{|\{n : a_n = \hat{a_n} + i\}|}{N} \right\}$$

All ILR scores in this paper were computed using a uniform pseudocount of 0.05.

7.1 Finding the optimal motif

For small datasets it is possible to employ a branch-and-bound algorithm for finding the motif with the optimal entropy (for a more elaborate approach addressing a similar problem see [4]). To see this, consider the space of alignments represented as a tree with the root representing the empty alignment and a node at depth n having $L - w + 1$ children corresponding to the choices of extending the alignment using the $(n + 1)^{\text{th}}$ sequence. A depth-first search (DFS) can then be employed on this tree to enumerate all the alignments at depth N and select the optimal motif. However, since complete enumeration is computationally expensive, even for very small datasets, we rely on pruning the search tree by not extending alignments that cannot possibly score better than the best score (s_{max}) that we have seen till now. This determination is made based on the following lemma:

LEMMA: *Let c_n denote the nucleotide counts for an alignment column of n sequences. Then*

$$\max_{c^N \ge c^n} I(c^N) = \max_{a \in \{A,C,G,T\}} I(c^n + (N - n)\delta_a),$$

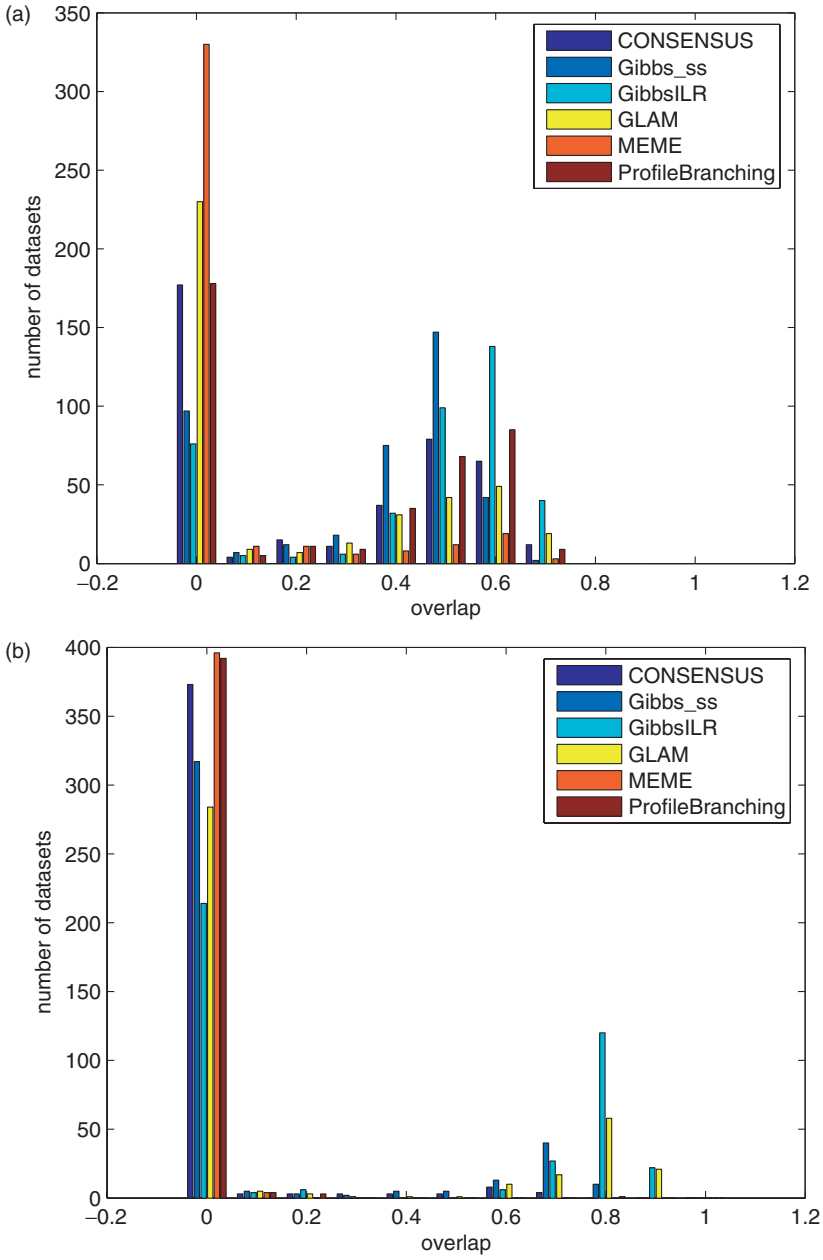


Fig. 5. Histogram of the number of datasets as a function of the amount of overlap with the implanted motif. (a) COMBO experiment (b) FIFTY experiment.

where δ_a *has a count of* 1 *for a and* 0 *otherwise*, $x \geq y$ *denotes pointwise inequalities*, $I_j(x) = x \log (x/\Theta_{0_j})$ *and* $I(c^n) = (\sum_j I_j(c_j^n)) - n \log n$ *is the entropy for a single column.*

PROOF OUTLINE: Suppose that there exists a maximally scored $c^N \geq c^n$ that is not of the form $c^n + (N-n)\delta_a$. Let $j = \text{argmax}_l I_l(c_l^N + 1) - I_l(c_l^N)$ and let $k \neq j$ be such that $c_k^N > c_k^n$ (there must exist such a k by definition of c^N). Then, since $I_l(x)$ is a monotonically increasing function, it can be shown that $I(c^N + \delta_j - \delta_k) > I(c^N)$, giving rise to a contradiction.

In words, the lemma says that an alignment column can be optimally extended in only four different ways and so we can quickly compute the optimal score that can arise out of the extension of a given alignment. In practice, this pruning strategy reduces the search space dramatically and allows us to find optimal motifs for moderate sample sizes (e.g. $w = 7$, $N = 10$ and $L = 100$). Note that we also provide the branch-and-bound algorithm with a good lower bound for s_{max}, as obtained from a motif-finding program such as CONSENSUS, to improve the initial pruning process and thus speed up the algorithm.

ACKNOWLEDGEMENTS

The last two authors are indebted to Pavel Pevzner for getting us hooked on these problems and for many discussions on the subject. This research uses computational resources funded by NIH grant 1S10RR020889. The last author would like to thank

the participants of the Second Barbados Workshop on Genomics and Gene Regulation hosted by McGill University Bellairs Research Institute in April 2005 for many interesting discussions. Finally, we wish to thank the anonymous referees for their valuable comments and suggestions.

REFERENCES

[1] Bailey,T.L. and Elkan,C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, Menlo Park, California, pp. 28–36.

[2] Bailey,T.L. and Elkan,C. (1995) The value of prior knowledge in discovering motifs with MEME. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 3, 21–29.

[3] Dempster,A.P. Laird,N.M. and Rubin,D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society, Series B*, 34, 1–38,

[4] Eskin,E. (2004) From profiles to patterns and back again: A branch and bound algorithm for finding near optimal motif profiles. In *Proceedings of the Eight Annual International Conference on Research in Computational Molecular Biology, RECOMB 2004, San Diego, USA*.

[5] Martin C Frith, Ulla Hansen, John L Spouge, and Zhiping Weng. (2004) Finding functional sequence elements by multiple local alignment. *Nucleic Acids Res.*, 32, 189–200.

[6] Hertz,G.Z. and Stormo,G.D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7–8), 563–577.

[7] Hughes,J.D., Estep,P.W., Tavazoie,S. and Church,G.M. (2000) Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. *J Mol Biol*, 296(5):1205–14.

[8] Jianjun Hu, Bin Li, and Daisuke Kihara. (2005) Limitations and potentials of current motif discovery algorithms. *Nucl. Acids Res.*, 33, 4899–4913.

[9] S. Karlin and S.F. Altschul. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc Natl Acad Sci USA*, 87(6), 2264–2268

[10] U Keich and PA Pevzner. (2002) Subtle motifs: defining the limits of motif finding algorithms. *Bioinformatics*, 18(10), 1382–1390.

[11] CE Lawrence, SF Altschul, MS Boguski, JS Liu, AF Neuwald, and JC Wootton. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262(5131), 208–14.

[12] Niranjan Nagarajan, Neil Jones, and Uri Keich. (2005) Computing the P-value of the information content from an alignment of multiple sequences. *Bioinformatics*, 21 Suppl 1(ISMB 2005):i311–i318.

[13] AF Neuwald, JS Liu, and CE Lawrence. (1995) Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein Sci*, 4(8), 1618–1632.

[14] P.A. Pevzner and S.H. Sze. (2000) Combinatorial approaches to finding subtle signals in DNA sequences. *Proc Int Conf Intell Syst Mol Biol*, 8:269–78.

[15] Prakash,A. and Tompa,M. (2005) Statistics of local multiple alignments. *Bioinformatics*, 21 Suppl 1:i344–50.

[16] Alkes Price, Sriram Ramabhadran, and Pavel A. Pevzner. Finding subtle motifs by branching from sample strings. *Bioinformatics*, 19(90002):149ii–155, 2003.

[17] Stormo,G.D. (2000) DNA binding sites: representation and discovery. *Bioinformatics*, 16(1), 16–23.

[18] Swets,J.A. (1988) Measuring the accuracy of diagnostic systems. *Science*, 240, 1285–1293.

[19] Martin Tompa, Nan Li, Timothy L Bailey, George M Church, Bart De Moor, Eleazar Eskin, Alexander V Favorov, Martin C Frith, Yutao Fu, W James Kent, Vsevolod J Makeev, Andrei A Mironov, William Stafford Noble, Giulio Pavesi, Graziano Pesole, Mireille Régnier, Nicolas Simonis, Saurabh Sinha, Gert Thijs, Jacques van Helden, Mathias Vandenbogaert, Zhiping Weng, Christopher Workman, Chun Ye, and Zhou Zhu. *et al*. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol*, 23(1), 137–44.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e402–e407
doi:10.1093/bioinformatics/btl258

Create and assess protein networks through molecular characteristics of individual proteins

Yanay Ofran[1,2,*,†], Guy Yachdav[1,2,3,†], Eyal Mozes[2], Ta-tsen Soong[2,4], Rajesh Nair[1,2] and Burkhard Rost[1,2,3]

[1]Department of Biochemistry and Molecular Biophysics, Columbia University, 650 West 168th Street, New York, NY 10032, USA, [2]Columbia University Center for Computational Biology and Bioinformatics (C2B2), 1130 St. Nicholas Ave. Rm. 802, New York, NY 10032, USA, [3]NorthEast Structural Genomics Consortium (NESG), Columbia University, 1130 St. Nicholas Ave. Rm. 802, New York, NY 10032, USA and [4]Department of Biomedical Informatics, Columbia University, 630 West 168th Street, New York, NY 10032, USA

ABSTRACT

Motivation: The study of biological systems, pathways and processes relies increasingly on analyses of networks. Most often, such analyses focus on network topology, thereby treating all proteins or genes as identical, featureless nodes. Integrating molecular data and insights about the qualities of individual proteins into the analysis may enhance our ability to decipher biological pathways and processes.

Results: Here, we introduce a novel platform for data integration that generates networks on the macro system-level, analyzes the molecular characteristics of each protein on the micro level, and then combines the two levels by using the molecular characteristics to assess networks. It also annotates the function and subcellular localization of each protein and displays the process on an image of a cell, rendering each protein in its respective cellular compartment. By thus visualizing the network in a cellular context we are able to analyze pathways and processes in a novel way. As an example, we use the system to analyze proteins implicated with Alzheimers disease and show how the integrated view corroborates previous observations and how it helps in the formulation of new hypotheses regarding the molecular underpinnings of the disease.

Availability: http://www.rostlab.org/services/pinat

Contact: pinat@rostlab.org; ofran@cubic.bioc.columbia.edu

1 INTRODUCTION

Protein-protein interaction (PPI) networks are believed to constitute a valuable framework for the analysis of biological processes. Several studies attempt to characterize the topological properties of PPI networks as a whole (Barabasi and Oltvai, 2004), or of small, recurring elements within them (Wuchty, *et al*., 2003). The biological implications of such topological observations are still debated (Bork, *et al*., 2004). However, it has been suggested that the analysis of PPI networks can help identify biological ''modules'' namely networks of a limited number of proteins that interact to carry out a certain process or function (Ge, *et al*., 2003; Hartwell, *et al*., 1999). Parsing the topology of such networks could help decipher biological processes and assign function to un-annotated proteins that are implicated in these modules (Vazquez, *et al*., 2003).

The first step in the topological analysis of modules is the generation of PPI networks from pairwise protein-protein interactions. Numerous databases curate and sometime even predict protein-protein interactions based on various criteria (Bader, *et al*., 2001; Hermjakob, *et al*., 2004; Peri, *et al*., 2003; Rhodes, *et al*., 2005; von Mering, *et al*., 2005; Xenarios, *et al*., 2002; Zanzoni, *et al*., 2002). Although the vast majority of these data come from high-throughput experiments, they also include manually curated data from the literature. High-throughput PPI data are often rather noisy, and include a substantial amount of false positives (Cusick, *et al*., 2005). In particular, yeast two-hybrid experiments (Y2H) can yield false positive results of two kinds. (1) Experimental errors: two proteins observed to physically bind, may not interact in reality. (2) ''*In vitro*'' error: the conditions under which Y2H experiments are carried out may lead to interactions that do not occur *in vivo*. While the first type of errors can be reduced substantially by rather simple experimental adjustments, the second type of error is harder to control. The most effective approach thus far for identifying these false positives on a large-scale is through *in silico* analysis (Cusick, *et al*., 2005). Problems with the reliability or reproducibility of data are not confined to high throughput PPI dataset. A comparison of several datasets that were collected by experts from the literature revealed that the overlap between such sets is small (Ramani, *et al*., 2005), calling for caution when using them in an automatic manner. Thus, when using these data, it is imperative to assess the reliability of specific interactions.

Another problem with the analysis of PPI networks relates to data representation. Many higher-level studies of biological networks treat individual proteins as featureless nodes and focus their analysis on the topology of network graphs. Yet, the molecular details of the individual proteins are crucial for understanding and assessing networks. There is an essential connection between the structure of a PPI network and the molecular features of each protein. For example, most eukaryotic proteins are confined to particular subcellular compartments. Biological processes that span different compartments often consist of several modules. Each of these modules is typically localized to a different compartment, and a small number of proteins serve as connectors between compartments. The localization of a protein is instrumental for assessing PPI data, as proteins that reside in different compartments are less

*To whom correspondence should be addressed.
†Equal contribution.

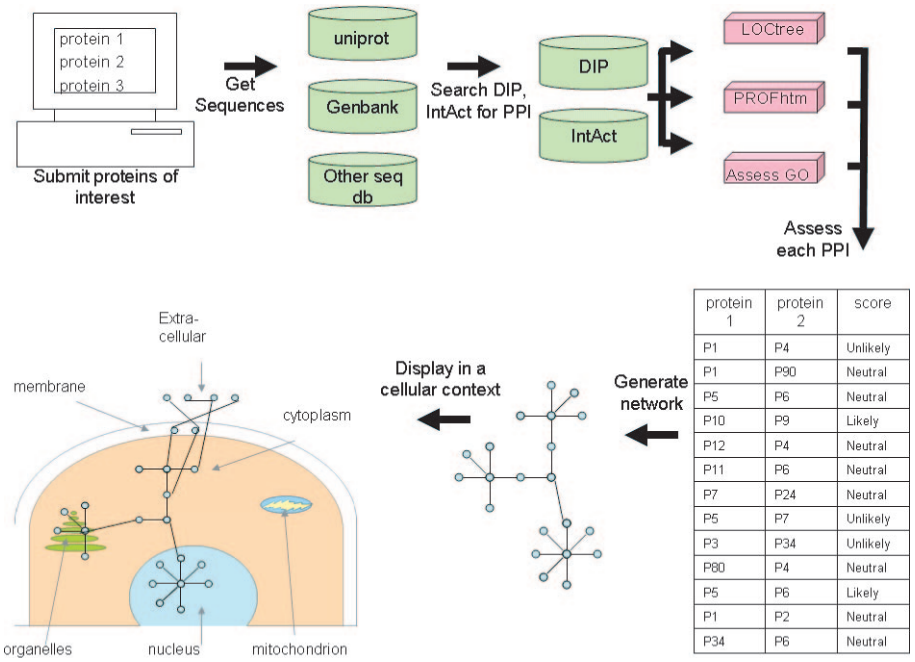


Fig. 1. Flow of the analysis and integration in PiNat. PiNAT accepts protein names as input. Its output is the PPI network around these proteins, rendered on a figure of a cell representing the interplay between the different cellular compartments. It also returns a table with a score for the biological likelihood of each interaction. This is done by integrating molecular data regarding the individual proteins, PPI data and systems view of the process. In particular, using analysis of GO annotation and of predicted SCL PiNAT grades the interactions according to their biological likelihood and renders them at the appropriate SCL.

likely to interact than proteins that are located to the same compartment (Sprinzak, *et al*., 2003; von Mering, *et al*., 2002). Similarly, proteins with incompatible functional annotation are assumed not to be very likely to interact (Sprinzak, *et al*., 2003). It is increasingly acknowledged that there is a need for a framework that will integrate the micro level, namely the characteristics of each protein such as its localization and functional annotation, with the macro level, namely the topology of the network. A system of representation and analysis that will offer such integration may improve our ability to elicit reliable and useful insights from high-throughput PPI data.

Here we introduce PiNAT (Protein interaction Network Assessment Tool)—an automated system that generates PPI networks around proteins of interest. It automatically analyzes each sequence, assesses the reliability of the interactions based on molecular criteria, and displays the network within an image of a cell, in a way that represents the flow of the process between its compartments. The automatic assessment is based on the results of a large-scale meticulous analysis of a large, highly reliable dataset of PPI. Once a list of proteins is submitted to the system, the following sequence of events is initiated (Fig. 1):

(1) PiNAT automatically queries databases of PPIs and constructs a network of known PPIs.

(2) The sequences of all proteins are obtained.

(3) Based on these sequences the system predicts the subcellular localization for all proteins, including proteins that have no homologue with experimental annotations about localization.

(4) Each interaction is graded using a likelihood based on the predicted localization of the participating proteins.

(5) Where available, the GO annotation of each protein is obtained.

(6) Each interaction is graded based on the likelihood of interaction between proteins with these annotations.

(7) Finally, the network of interactions is displayed in a cellular context. It is readily visible from this display how the process flows between the different compartments of the cell.

We demonstrate the power of the system by generating, assessing and displaying the known fraction of the PPI network that underlies Alzheimer's disease (AD).

2 METHODS

2.1 Large-scale assessment of PPIs based on localization

We used the DIP core dataset (Deane, *et al*., 2002; Xenarios, *et al*., 2002) to generate the localization-based scores for PPIs. This is a large, reliable set of interactions each of which was observed by at least three different methods. We predicted the localization for each protein in this dataset and checked the probability of observing interactions between proteins from any combination of localizations.

Subcellular localization was predicted using two methods: (1) LOCtree, (Nair and Rost, 2005) that assigns the following major classes to eukaryotic proteins: extra-cellular space, cytoplasm, organelles, mitochondrion or nucleus. (2) PHDhtm, (Rost, *et al*., 1996) that predicts transmembrane helices. The sustained performance of both methods has been thoroughly established. LOCtree assigns each prediction a confidence level between 1 (low) and 10 (high). We considered LOCtree predictions with confidence scores <4 as ''low confidence'' and discarded them from the assessment. PHDhtm predicts whether or not a certain residue is embedded in a transmembrane

helix (TMH) and assigns the prediction a confidence level between 0 (low) and 9 (high). We deemed a protein transmembrane if: (a) PHDhtm identified at least 20 transmembrane residues, and (b) the average confidence score of the 20 most reliable predictions was above 8.5. In the gold-standard analysis described below, we found that with these thresholds about 7% of the proteins were identified as transmembrane and approximately 60% of the nodes were predicted in a particular localization with high confidence. All other proteins were designated unknown.

For each pair of subcellular compartments, we assigned a likelihood grade of ''likely'', ''unlikely'' or ''neutral'', indicating the likelihood of interaction among proteins from these compartments. These grades were determined as follows: We ran LOCtree and PHDhtm for 4800 interactions from the DIP core set, involving a total of 2191 proteins. 1482 of the 2191 proteins were given a high-confidence prediction by either PHDhtm or LOCtree. Of the 4800 interactions, 2312 had high-confidence predictions for both proteins.

Since we had a total of 1482 proteins with high-confidence predictions, the total number of protein pairs—assuming symmetry—was 1,097,421; of these, 2312 ($\sim$1/475) were well-documented interactions. If we take as our null hypothesis that the knowledge of localization has no effect on the probability of interaction, the approximate expected number of well-documented interactions for each pair of compartments will be the total number of PPIs in this pair of compartments divided by 475. For each pair of compartments, we determined whether it was over- or under represented in the subset of well-documented interactions. We then used the binomial approximation to the cumulative hypergeometric probability distribution, to assign a p-value to this over- or under-representation. We used a p-value threshold of 0.01, i.e. we assigned each pair of categories a likelihood grade of ''likely'' if it was over-represented with a p-value <0.01; a grade of ''low'' if it was under-represented with a p-value <0.01; and a grade of ''neutral'' otherwise.

When analyzing a network, each edge is assigned a likelihood grade based on the predicted compartments of its two nodes. An assessment of likely or unlikely is assigned to an edge only if we have high-confidence predictions, from either PHDhtm or LOCtree, for both nodes. If one or both of the nodes have only low-confidence predictions, the edge is always assigned a neutral grade.

2.2 Automatic generation of networks

The first stage of the analysis in PiNAT is the automatic generation of a PPI network. This is done by taking the list of protein names submitted by the user and search both DIP (Xenarios, *et al*., 2002) and IntAct (Hermjakob, *et al*., 2004) for the interactions involving them. Users can specify what depth of the interaction tree around the proteins they are interested in. For example, a depth of 1 will retrieve all the proteins that interact with any of the query proteins; a depth of 2 will retrieve also the proteins that interact with the proteins at depth 1, and so forth. Finally, based on the protein names and accession numbers the sequences are retrieved from the relevant sequence database. It is also optional to submit to the PiNAT server a list of sequences or a complete interaction network.

2.3 Large-scale assessment of PPIs based on GO

Proteins in one biological process are more likely to interact than proteins in distinct processes. Therefore, we used the GO annotations of each protein in order to grade the likelihood of interaction between them. Since GO includes records inferred electronically (i.e. based on sequence or structure similarity), we only take the annotations that come from trusted experiments such as direct assays. We measured the distance between two GO terms as the information content of the minimum subsumer of the two terms (Lord, *et al*., 2003). Low information content reflects a highly specific concept shared by the two terms and indicates a close relationship between them. Since there is often more than one GO annotation available for a particular protein, for every annotation c_k in protein i, we found its most similar term $c_{j\max}$ in protein j, and vice versa for each annotation in protein j. We then averaged these best similarities to obtain the GO score between the two proteins (Eqn. 1).

$$similarity\ (i,\ j) = \frac{\sum_{k=1}^{m} simGO\ (c_k, c_{j\max}) + \sum_{p=1}^{n} simGO\ (c_{i\max}, c_p)}{m+n} \quad \text{Eqn.(1)}$$

where m and n are the respective numbers of annotations in i and j, and $simGO(c_A, c_B)$ is the GO similarity between terms c_A and c_B according to the definition by Lord *et al*.

We used the proteins in the DIP core set and generated 100,000 random pairings of these proteins to derive a background distribution of GO similarity scores.

2.4 Display of networks in the cellular context

The predictions from LOCtree and PHDhtm are also used to visualize the location of the nodes in the network drawing in the following manner. Given the network and the predictions from LOCtree and PHDhtm, we generate a Graph Markup Language (GML) file for Cytoscape (Shannon, *et al*., 2003), placing each node in the drawing according to its predicted localization. Nodes in the drawing are divided among six groups: one for each of LOCtree's five categories, and one for membranes. Note that we, incorrectly, assumed that all membrane proteins reside in the cytoplasmic membrane, due to the lack of accurate method that distinguishes *in silico* between proteins in different membranes. For purposes of placing a node in the drawing, we used the following intuition-based rules: a high-confidence prediction from PHDhtm overrides a high-confidence prediction from LOCtree; a high-confidence prediction from LOCtree overrides a low-confidence prediction from PHDhtm; and a low-confidence prediction from PHDhtm overrides a low-confidence prediction from LOCtree. There is also a seventh group for nodes for which LOCtree was unable to give even a low-confidence prediction; such cases, however, are relatively rare (<1%).

2.5 Alzheimer's disease related pathway

A pathway of proteins implicated in Alzheimer was retrieved from the KEGG database (Goto, *et al*., 1997). The pathway includes 21 proteins and was manually gleaned from the literature. We used the 21 proteins as input to the PiNAT server, composed the network of interactions around them (depth=1), identified the most likely and the most unlikely interactions and rendered the network in a cellular context.

3 RESULTS AND DISCUSSION

3.1 Interactions across subcellular compartments

A first glance at the results of the large-scale assessment of PPIs based on subcellular localization (Table 1) confirmed the intuition: almost always, proteins from the same compartment had a higher chance of interacting with each other than do pairs of proteins from different compartments. The exceptions for the intra-compartment interactions were extracellular proteins that did not show a significant tendency to interact with each other. This makes biological sense, as extracellular proteins are often messengers that facilitate communication between cells. Hence, it is not surprising to find that they show only weak interaction preferences. In contrast, almost all low scores originated from distant compartments. Conversely, PPIs between nearby compartments were found to be likely. An exception to this was the interaction between transmembrane and cytoplasmic proteins, and between transmembrane and organellar proteins. While the first (transmembrane-cytoplasmic) was significantly lower than random, the latter (transmembrane-organellar) was significantly higher. This could be explained by the intricate trafficking system of

Table 1. scores for interactions between compartments

	Extra cellular	Cytoplasmm	Orgnl	Mitochondrion	Nuclear	TM
Extra cellular	Neutral					
Cytoplasmic	Neutral	High				
Orgnnellar	Neutral	Neutral	High			
Mitochondrial	Neutral	Low	Neutral	High		
Nuclear	Low	High	Low	Low	High	
TM	Neutral	Low	High	Neutral	Low	High

For each combination of subcellular compartments we computed whether the interaction between proteins from these compartments has a high probability, is neutral or has a low probability. The calculation is based on a large dataset of reliable PPI. Low, high significance refers to the expected probability under the null hypothesis (H_0: localization has no effect on the probability of interaction) with p-values <0.01 (for under- or over-representation). Combinations that did not differ significantly from the expectation were deemed neutral.

proteins to the membrane, which involves the organelles: globular proteins that interact with membrane proteins often get to their destination through the secretory pathway rather than through free diffusion in the cytoplasm.

3.2 Likely and unlikely interactions across GO

Fig. 2 shows the scores obtained from the analysis of GO annotations for positive interactions (i.e. interactions included in DIP core set) and negative interaction (randomly paired proteins from the core set). We found that over 52% of the positive interactions get a score higher than 3.25 while 95% of the negative ones get a score lower than 3.25. When using a lower threshold of 1.3, we could retain 81% of the positives but only reject half of the negatives. We therefore binned the GO similarity scores into three categories: ''likely'' for interactions with score greater than 3.25, ''unlikely'' for interactions with score smaller than 1.3, and ''neutral'' for any score within that range. Note, that since the ratio of interacting pairs to non- interacting pairs in a proteome is very small (Grigoriev, 2003), even at this cutoff false positive interactions will outnumber true positives. However, most of the negative interactions will be rejected while most of the positive ones will be accepted.

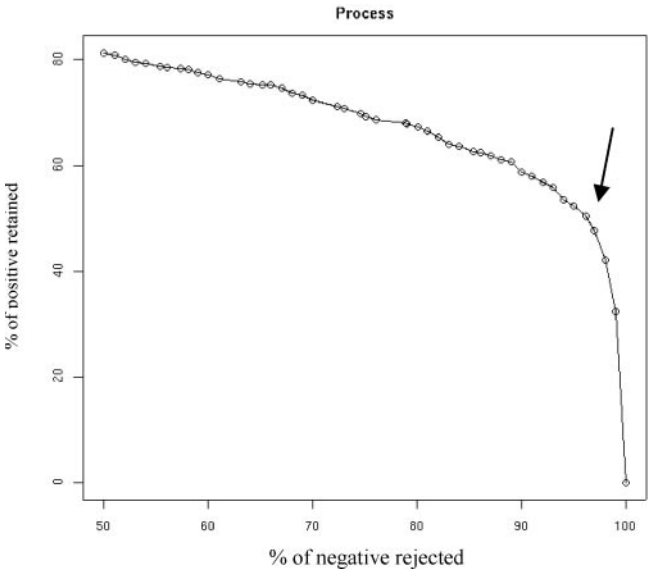


Fig. 2. Scores of interactions according to GO annotations on positive and negative data. Each point on the graph represents a certain score for interaction between proteins with different GO annotations. On the x axis is the percentage of negative (i.e. random) interaction that are below that score. On the y axis is the percentage of real PPI that get a score above that score. For a score 3.25 (arrow) 95% of the negative interactions will be rejected and the 52.4% of the positive ones will be retained.

3.3 Alzheimer in the perspective of PiNAT

The main pathological manifestations in Alzheimer's are neuritic plaques and neurofibrillary tangles, both are abnormal protein aggregations. They differ in the proteins that are accumulated in them and in their localization. While the first occur in the extracellular space, the latter occur around the cytoskeleton in the cytoplasm (Chapman, *et al*., 2001). All the genes that were found to be linked to the disease are involved in the production or deposition of these aggregations (Selkoe, 2001). The mapping of compartments to the PPI network that is implicated in this process is, therefore, of particular interest. Fig. 3 shows the Alzheimer's related PPI, rendered by PiNAT into a cellular context. The main hub in this network is the Amyloid beta A4 protein (APP), a derivative of which constitutes the neuritic plaques. The function of APP is not entirely clear. It is generally accepted that it is a cell surface receptor (Selkoe, 2001).

However, it has been shown that APP is often cleaved and that part of it is secreted (Selkoe, 2001). It has also been reported that derivatives of APP were observed in the cytoplasm (Selkoe, 2001). APP can promote transcription activation, and it has been reported that some derivatives of it are located in the nucleus (Selkoe, 2001). The figure reflects this unclarity regarding APP's localization. It is displayed in the nucleus according to the LOCtree prediction (the prediction of LOCtree was given a confidence score of 4, which is our lower bound for accepting LOCtree predictions). However, PHDhtm identified a short transmembrane segment that due to its length fell just below the cutoff we set for considering proteins as transmembrane (Methods). Yet, the pattern of interactions around APP is in agreement with all the reported observations regarding its localization. APP interacts extensively with almost every compartment of the cell. For example, there are 17 proteins that are predicted to be nuclear in this network. Most of them are part of a connected component. However, if APP is removed from the network, the connected component immediately disintegrates leaving only two of the nuclear proteins connected to each other. Hence, PiNAT's display of the network corroborates the findings regarding APP's location.

Thirteen of the Alzheimer PPIs were deemed unlikely according to their localization (Table 2). Most of these interactions involved

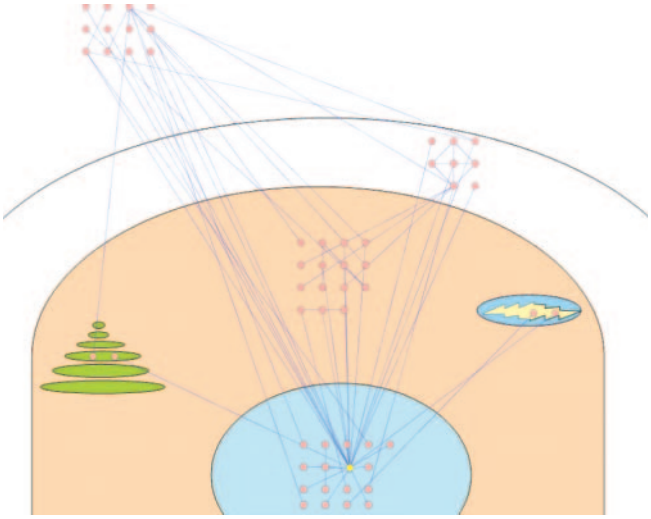


Fig. 3. Output of PiNAT for Alzheimer's disease-related PPI network. The proteins known to be involved in a pathway related to AD were used as an input to PiNat. Their PPIs were collected from DIP and IntAct to generate a network. Each protein in the network is displayed in its SCL according to the predictions of LOCtree and PHDhtm. The major hub of this network is the APP protein (colored yellow), whose metabolism is a major key for understanding the pathologies of AD. The myriad of interactions it has with many compartments of the cell is in agreement with suggestions regarding its functional diversity.

APP. This is understandable, as many interactions between APP, which was predicted to be nuclear, and proteins in non-nuclear compartments are considered unlikely. Still, knowing that APP and its derivatives can reside in different compartments calls for reevaluation of this classification. By and large, the two scoring schemes were in agreement about the likelihood of most interactions. The two scoring schemes radically disagreed for only one PPI, namely the interaction between NEDD8 Amyloid protein binding protein (ULA1_HUMAN) and APP (A4_HUMAN). The GO annotations of these proteins were very different. Hence their GO derived interaction score was low. However, since they were predicted to be in spatial proximity, the localization-based method scored the interaction between them as highly likely. This disagreement could also be ascribed to our poor understanding of APP and its function.

Interestingly, several of the proteins in this network have little or no functional annotation. Viewing them in the context of the cellular process can help to postulate hypotheses regarding the role they may have in Alzheimer's. It is widely accepted that inhibiting the cleavage of APP can slow down the advance of the disease and may even help prevent it. Thus, the exact localization in which the cleavage takes place is of a great interest. Identifying proteins in the network that may be involved in this cleavage may offer some important insights into this problem. Careful analysis of the localized network can suggest some additional insights into the molecular underpinnings of Alzheimer's disease and even help formulate new hypotheses. For example, it may be possible to determine which of the un-annotated proteins in the network may be involved in cleaving APP. Their SCL could serve to identify where the cleavage occurs. Clearly, such analysis is beyond the scope of this paper.

Table 2. Scores for Alzheimer related interactions

protein 1 (UniProt)	protein 2 (UniProt)	SCL score	GO derived score
A4_HUMAN	A2MG_HUMAN	UNLIKELY	NEUT
A4_HUMAN	A4_HUMAN	LIKELY	NEUT
A4_HUMAN	ABB1_HUMAN	LIKELY	NEUT
A4_HUMAN	ABB2_HUMAN	LIKELY	NEUT
A4_HUMAN	ABB3_HUMAN	LIKELY	NEUT
A4_HUMAN	ACES_HUMAN	NEUT	UNLIKELY
A4_HUMAN	ACH7_HUMAN	UNLIKELY	NEUT
A4_HUMAN	APA1_HUMAN	UNLIKELY	NEUT
A4_HUMAN	APB1_HUMAN	NEUT	NEUT
A4_HUMAN	APE_HUMAN	UNLIKELY	NEUT
A4_HUMAN	ASP2_HUMAN	LIKELY	NEUT
A4_HUMAN	BACE1_HUMAN	UNLIKELY	UNLIKELY
A4_HUMAN	HCD2_HUMAN	UNLIKELY	UNLIKELY
A4_HUMAN	JIP1_HUMAN	LIKELY	NEUT
A4_HUMAN	LRP1_HUMAN	NEUT	UNLIKELY
A4_HUMAN	Q9UCX5	LIKELY	NEUT
A4_HUMAN	SHC1_HUMAN	NEUT	UNLIKELY
A4_HUMAN	TGF1_HUMAN	UNLIKELY	NEUT
A4_HUMAN	TGFB2_HUMAN	UNLIKELY	NEUT
A4_HUMAN	TTHY_HUMAN	UNLIKELY	NEUT
A4_HUMAN	ULA1_HUMAN	LIKELY	UNLIKELY
ABB3_HUMAN	A4_HUMAN	LIKELY	NEUT
APH1A_HUMAN	PSN1_HUMAN	LIKELY	LIKELY
BIR2_HUMAN	CASP3_HUMAN	NEUT	NEUT
BIR2_HUMAN	CASP7_HUMAN	NEUT	NEUT
CASP3_HUMAN	BIR7_HUMAN	NEUT	NEUT
FLNA_HUMAN	PSN1_HUMAN	NEUT	NEUT
G3P2_HUMAN	A4_HUMAN	LIKELY	NEUT
GNB:3712673	PSN1_HUMAN	UNLIKELY	NEUT
GSK3B_HUMA	REN3A_HUMAN	LIKELY	NEUT
IF38_HUMAN	NEP_HUMAN	NEUT	NEUT
LIPL_HUMAN	ACC2_HUMAN	NEUT	NEUT
LIPL_HUMAN	CSN6_HUMAN	NEUT	NEUT
LIPL_HUMAN	L7L2_HUMAN	NEUT	NEUT
LIPL_HUMAN	LRP1_HUMAN	NEUT	UNLIKELY
LIPL_HUMAN	PTN4_HUMAN	NEUT	UNLIKELY
LIPL_HUMAN	Q7L354	NEUT	NEUT
LIPL_HUMAN	Q9P2H0	NEUT	NEUT
LIPL_HUMAN	RL18A_HUMAN	NEUT	UNLIKELY
LIPL_HUMAN	ULA1_HUMAN	NEUT	UNLIKELY
LRP1_HUMAN	A2MG_HUMAN	NEUT	NEUT
NICA_HUMAN	APH1A_HUMAN	LIKELY	LIKELY
NICA_HUMAN	PSN1_HUMAN	LIKELY	LIKELY
O00193	A2MG_HUMAN	UNLIKELY	NEUT
PEN2_HUMAN	APH1A_HUMAN	LIKELY	LIKELY
PEN2_HUMAN	NICA_HUMAN	LIKELY	LIKELY
PEN2_HUMAN	PSN1_HUMAN	LIKELY	LIKELY
Q7Z4Y5	A2MG_HUMAN	NEUT	NEUT
Q9H5B5	TAU_HUMAN	NEUT	NEUT
Q9UJZ5	PSN1_HUMAN	NEUT	NEUT
R11A_HUMAN	PSN1_HUMAN	UNLIKELY	NEUT
RL10_HUMAN	PSN1_HUMAN	NEUT	NEUT
TAU_HUMAN	TBBX_HUMAN	NEUT	UNLIKELY

List of PPI that were extracted from DIP and IntAct to generate the Alzheimer's related network. In the first and second columns are the UniProt protein names for each of the proteins in a given interaction. The third column is the SCL-based score for this interaction and the fourth column is the GO derived score

4 CONCLUSIONS

The integration of molecular knowledge and network structure can enhance our understanding of biological processes and of pathways. PiNAT, which is fully automated, offers a framework for combining the micro-level analysis of individual molecules and the macro-level of network topology. Users can submit a single protein, a list of proteins, or a whole network as an input. As an output they will receive a visual description of the predicted spatial flow of a pathway in the cell. In addition the user will get a list of scores for each interaction, based on different sequence-level analyses of the individual proteins. PiNAT is easily expandable. Thus, it will be possible to add many other molecular and network analyses to improve our insights into pathways and modules. Our example for the case of Alzheimer's illustrated just some aspects of the usefulness of PiNAT. The server is available at: www.rostlab.org/services/pinat

5 ACKNOWLEDGEMENTS

Thanks to Kazimierz O. Wrzeszczynski (Columbia) for helpful comments and discussion. This work was supported by the grants: RO1-GM64633-01 from the National Institutes of Health (NIH) and the grant RO1-LM07329-01 from the National Library of Medicine (NLM). Last, not least, thanks to all those who deposit their experimental data in public databases, and to those who maintain these databases.

6 REFERENCES

Bader,G.D., Donaldson,I., Wolting,C., Ouellette,B.F., Pawson,T. and Hogue,C.W. (2001) BIND—The Biomolecular Interaction Network Database. *Nucleic Acids Res*, 29, 242–245.

Barabasi,A.L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat Rev Genet*, 5, 101–113.

Bork,P., Jensen,L.J., von Mering,C., Ramani,A.K., Lee,I. and Marcotte,E.M. (2004) Protein interaction networks from yeast to human. *Curr Opin Struct Biol*, 14, 292–299.

Chapman,P.F., Falinska,A.M., Knevett,S.G. and Ramsay,M.F. (2001) Genes, models and Alzheimer's disease. *Trends Genet*, 17, 254–261.

Cusick,M.E., Klitgord,N., Vidal,M. and Hill,D.E. (2005) Interactome: gateway into systems biology. *Hum Mol Genet*, 14 Spec No. 2, R171–181.

Deane,C.M., Salwinski,L., Xenarios,I. and Eisenberg,D. (2002) Protein interactions: two methods for assessment of the reliability of high throughput observations. *Mol Cell Proteomics*, 1, 349–356.

Ge,H., Walhout,A.J. and Vidal,M. (2003) Integrating 'omic' information: a bridge between genomics and systems biology. *Trends Genet*, 19, 551–560.

Goto,S., Bono,H., Ogata,H., Fujibuchi,W., Nishioka,T., Sato,K. and Kanehisa,M. (1997) Organizing and computing metabolic pathway data in terms of binary relations. *Pac Symp Biocomput*, 175–186.

Grigoriev,A. (2003) On the number of protein-protein interactions in the yeast proteome. *Nucleic Acids Res*, 31, 4157–4161.

Hartwell,L.H., Hopfield,J.J., Leibler,S. and Murray,A.W. (1999) From molecular to modular cell biology. *Nature*, 402, C47–52.

Hermjakob,H., Montecchi-Palazzi,L., Lewington,C., Mudali,S., Kerrien,S., Orchard,S., Vingron,M., Roechert,B., Roepstorff,P., Valencia,A., Margalit,H., Armstrong,J., Bairoch,A., Cesareni,G., Sherman,D. and Apweiler,R. (2004) IntAct: an open source molecular interaction database. *Nucleic Acids Res*, 32, D452–455.

Lord,P.W., Stevens,R.D., Brass,A. and Goble,C.A. (2003) Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 19, 1275–1283.

Nair,R. and Rost,B. (2005) Mimicking cellular sorting improves prediction of subcellular localization. *J Mol Biol*, 348, 85–100.

Peri,S., Navarro,J.D., Amanchy,R., Kristiansen,T.Z., Jonnalagadda,C.K., Surendranath,V., Niranjan,V., Muthusamy,B., Gandhi,T.K., Gronborg,M., Ibarrola,N., Deshpande,N., Shanker,K., Shivashankar,H.N., Rashmi,B.P., Ramya,M.A., Zhao,Z., Chandrika,K.N., Padma,N., Harsha,H.C., Yatish,A.J., Kavitha,M.P., Menezes,M., Choudhury,D.R., Suresh,S., Ghosh,N., Saravana,R., Chandran,S., Krishna,S., Joy,M., Anand,S.K., Madavan,V., Joseph,A., Wong,G.W., Schiemann,W.P., Constantinescu,S.N., Huang,L., Khosravi-Far,R., Steen,H., Tewari,M., Ghaffari,S., Blobe,G.C., Dang,C.V., Garcia,J.G., Pevsner,J., Jensen,O.N., Roepstorff,P., Deshpande,K.S., Chinnaiyan,A.M., Hamosh,A., Chakravarti,A. and Pandey,A. (2003) Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Res*, 13, 2363–2371.

Ramani,A.K., Bunescu,R.C., Mooney,R.J. and Marcotte,E.M. (2005) Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. *Genome Biol*, 6, R40.

Rhodes,D.R., Tomlins,S.A., Varambally,S., Mahavisno,V., Barrette,T., Kalyana-Sundaram,S., Ghosh,D., Pandey,A. and Chinnaiyan,A.M. (2005) Probabilistic model of the human protein-protein interaction network. *Nat Biotechnol*, 23, 951–959.

Rost,B., Fariselli,P. and Casadio,R. (1996) Topology prediction for helical transmembrane proteins at 86% accuracy. *Protein Sci*, 5, 1704–1718.

Selkoe,D.J. (2001) Alzheimer's disease: genes, proteins, and therapy. *Physiol Rev*, 81, 741–766.

Shannon,P., Markiel,A., Ozier,O., Baliga,N.S., Wang,J.T., Ramage,D., Amin,N., Schwikowski,B. and Ideker,T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*, 13, 2498–2504.

Sprinzak,E., Sattath,S. and Margalit,H. (2003) How reliable are experimental protein-protein interaction data? *J Mol Biol*, 327, 919–923.

Vazquez,A., Flammini,A., Maritan,A. and Vespignani,A. (2003) Global protein function prediction from protein-protein interaction networks. *Nat Biotechnol*, 21, 697–700.

von Mering,C., Jensen,L.J., Snel,B., Hooper,S.D., Krupp,M., Foglierini,M., Jouffre,N., Huynen,M.A. and Bork,P. (2005) STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res*, 33, D433–437.

von Mering,C., Krause,R., Snel,B., Cornell,M., Oliver,S.G., Fields,S. and Bork,P. (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417, 399–403.

Wuchty,S., Oltvai,Z.N. and Barabasi,A.L. (2003) Evolutionary conservation of motif constituents in the yeast protein interaction network. *Nat Genet*, 35, 176–179.

Xenarios,I., Salwinski,L., Duan,X.J., Higney,P., Kim,S.M. and Eisenberg,D. (2002) DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*, 30, 303–305.

Zanzoni,A., Montecchi-Palazzi,L., Quondam,M., Ausiello,G., Helmer-Citterich,M. and Cesareni,G. (2002) MINT: a Molecular INTeraction database. *FEBS Lett*, 513, 135–140.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e408–e416
doi:10.1093/bioinformatics/btl222

BaCelLo: a balanced subcellular localization predictor

Andrea Pierleoni, Pier Luigi Martelli, Piero Fariselli and Rita Casadio*

Biocomputing Group, Dept. of Biology University of Bologna, via Irnerio 42, 40126 Bologna, Italy

ABSTRACT

Motivation. The knowledge of the subcellular localization of a protein is fundamental for elucidating its function. It is difficult to determine the subcellular location for eukaryotic cells with experimental high-throughput procedures. Computational procedures are then needed for annotating the subcellular location of proteins in large scale genomic projects.

Results. BaCelLo is a predictor for five classes of subcellular localization (secretory pathway, cytoplasm, nucleus, mitochondrion and chloroplast) and it is based on different SVMs organized in a decision tree. The system exploits the information derived from the residue sequence and from the evolutionary information contained in alignment profiles. It analyzes the whole sequence composition and the compositions of both the N- and C-termini. The training set is curated in order to avoid redundancy. For the first time a balancing procedure is introduced in order to mitigate the effect of biased training sets. Three kingdom-specific predictors are implemented: for animals, plants and fungi, respectively. When distributing the proteins from animals and fungi into four classes, accuracy of BaCelLo reach 74% and 76%, respectively; a score of 67% is obtained when proteins from plants are distributed into five classes. BaCelLo outperforms the other presently available methods for the same task and gives more balanced accuracy and coverage values for each class. We also predict the subcellular localization of five whole proteomes, *Homo sapiens*, *Mus musculus*, *Caenorhabditis elegans*, *Saccharomyces cerevisiae* and *Arabidopsis thaliana*, comparing the protein content in each different compartment.

Availability. BaCelLo can be accessed at http://www.biocomp.unibo.it/bacello/

Contact. casadio@alma.unibo.it, andrea@biocomp.unibo.it, gigi@biocomp.unibo.it, piero@biocomp.unibo.it

1 INTRODUCTION

The eukaryotic cell is a composite system internally subdivided into membrane-enveloped compartments that perform particular functions. Every subcellular compartment contains specific proteins, including enzymes, synthesized in the cytoplasm and translocated into the locations, where they carry out functional patterns. Therefore, knowing the localization of every protein is important for elucidating its interactions with other molecules and for understanding its biological function. Experimental high-throughput approaches have been applied to determine protein localization in *Saccharomyces cerevisiae* (Huh *et al*., 2003) and in *Arabidopsis thaliana* (Kleffmann *et al*., 2004). However these techniques cannot be generally applied to all the eukaryotic cells and computational predictive methods are needed in order to screen the huge amount of data derived from genomic projects and to guide the design of experiments.

Intracellular protein sorting involves several post-translational mechanisms that redirect a newly synthesized chain from the cytosol to its specific compartment on the basis of the information contained in its residue sequence. Pre-translational mechanisms, involving the sorting of the mRNAs inside the cytosol, seem to play a minor role in the translocation between different compartments (Gonsalvez *et al*., 2005). These considerations lead to the conclusion that the residue sequence of a protein is mostly responsible for its localization.

It is well known that many sequences contain cleavable peptides at the N-terminus that address the protein either to the secretory pathway, in which case they are called signal peptides, or to mitochondria and plastids and in this case they are called target or transit peptides. Some predictors have been implemented in order to recognize N-terminal signal peptides (Nielsen *et al*., 1997; Fariselli *et al*., 2003) or both the signal and the transit peptides (Emanuelsson *et al*., 2000). However some proteins get secreted by means of a non-classical way and do not require N-terminal signal peptides (Nickel, 2003; Bendtsen *et al*., 2004). Furthermore some proteins are translocated into mitochondria owing to a localization peptide at the C-terminal region (Lee *et al*., 1999; Izeta *et al*., 2003; Yamada *et al*., 2004). For these reasons in order to obtain an accurate prediction it is necessary to add information besides the N-terminal composition.

Concerning nuclear proteins, they have to span the nuclear membranes through the proteic Nuclear Pore Complexes. Several crossing mechanisms have been described, including free diffusion and mediated transport. Nuclear Localization Signals have been reported (Fried and Kutay, 2003) and approaches for finding them into a protein sequence were tested. However, the methods that incorporate only this type of information do not achieve satisfactory performance, probably due to the shortness and low specificity of the signals. Moreover only 30% of nuclear proteins is estimated to have a NLS (Cokol *et al*., 2000).

A number of different predictors for the subcellular localization have been released in the past years, based on different approaches. They can be divided into two major classes, following Nair and Rost (2005): predictors based on the knowledge extracted from the annotated databases and the so called *de novo* predictors. The former ones are based on the detection of similarity between the sequence to be predicted and sequences with known localization, by searching for homology (Marcotte *et al*., 2000) or for conserved domains or motifs (Scott *et al*., 2004). However these tools are able to predict

*To whom correspondence should be addressed.

the localization of half of the sequences in the data bases (Nair and Rost, 2005). The *de novo* methods, which we are interested in, are more general and rely only on the analysis of the residue sequence, without inferring the annotation from any known sequences. pTARGET (Guda and Subramaniam, 2005) and ESLpred (Bhasin and Raghava, 2004) are hybrid methods since they take into consideration both the results of PFAM or BLAST and the analysis of the residue composition. Among the *de novo* methods, some exploit only the information contained in the N-terminal portions of the sequences, some consider the overall residue composition, and others take advantage of the evolutionary information contained in sequence profiles. They make use of both standard statistical methods and machine learning approaches, such as Neural Networks and Support Vector Machines. Moreover they discriminate different number of classes, from 3 up to 12. All the available predictors infer their parameters from unbalanced data sets: the size of different classes merely reflects the presence in the data bases of annotated sequences and it is unlikely that it can represent an estimate of the actual proportions in a living cell, as previously estimated at genomic level (Nair *et al*., 2005). This lead to an overestimation of the prediction for the most represented classes, namely the nuclear proteins for animal or fungi and the chloroplastic ones for plants.

Here we describe a novel method for subcellular localization prediction that adopts a balancing procedure by assuming a uniform *a priori* probability for the classes. The predictor makes use of several Support Vector Machines (SVMs), draws information from both the protein sequence and its profile derived with a BLAST search in the database of eukaryotic proteins, and considers in an explicit way the compositions of the whole sequence and of both the N- and C- termini.

Particular care has been taken in selecting the training set. Indeed most of the available methods, except LocTree (Nair and Rost, 2005), take into consideration proteins sharing high level of identity, up to 95%, and do not adopt rigorous validation procedures for excluding homology between the training and the testing sets. The justification for this procedure is that the subcellular localization of a protein can change owing to the change of few residues in the sequence, as in the case of the short nuclear localization signals. However, as we prove in this paper, in most practical cases, the high level of similarity between two sequences determines the same localization for the two proteins; in these cases a simple assignment based on the transfer of annotation after a BLAST search achieves a very good performance, even better than those reported by more sophisticated methods. Redundant data sets can therefore lead to methods that have a poor generalization capability. For these reason we select non-redundant data sets comprising proteins sharing less than 30% identity.

We consider five subcellular compartments: the secretory pathway, the cytoplasm, the nucleus, the mitochondrion and the chloroplast, when present. We decided not to predict more classes since for other locations the number of annotated non homologous protein chians is very low, not enough to train a predictor with a good generalization capability. We did not consider membrane proteins, since efficient methods for the prediction of transmembranicity are available, with very low rate of false positives and false negatives (about 3%, Martelli *et al*., 2003).

Differences concerning the localization mechanisms among the different kingdoms have been reported and we trained three different systems for animals, fungi and plants, respectively. In particular BaCelLo is the first *de novo* predictor that distinguishes between animal and fungal organisms. BaCelLo also takes advantage of the evolutionary information contained in sequence profiles that are known to improve performances of predictors for protein structure and function.

2 MATERIALS AND METHODS

2.1 Data sets

Starting from release 48 of the SWISS-PROT data base (Bairoch *et al*., 2005), we generated three data sets for animals (*Metazoa*), fungi (*Fungi*) and plants (*Viridiplantae*), respectively. Proteins with an experimental annotation of the subcellular location were retained, excluding those in which the comments 'fragment', 'possible', 'probable' and 'by similarity' are reported. We also excluded proteins with multiple subcellular localization and proteins shorter than 50 residues. The entries annotated as 'membrane'or 'transmembrane' were discarded, since we are interested only in globular proteins. The three data sets were separately clustered with an identity level equal to 30% using the BLASTCLUST tool and checked with BLASTp (Altschul *et al*., 1990). One representative protein for each cluster was selected. This procedure led to 2597 proteins from animals, 1198 proteins from fungi and 491 proteins from plants, distributed in five locations: nucleus, cytoplasm, secretory pathway (comprehending proteins annotated as 'Secretory' and 'Extracellular'), mitochondrion and chloroplast (Table 1a). Other subcellular localizations have been excluded because too few (less than 20) non-redundant representatives have been annotated so far.

Available predictors have been trained with data sets up to the release 41 of SWISS-PROT. For sake of comparison, we reduced our training sets extracting only the non-redundant proteins contained in that release. The remaining proteins have been used as independent test sets (Table 1b). Since the number of proteins in the plant test set is small, we don't report the results. All the data sets are available at www.biocomp.unibo.it/bacello

For predicting the localization of proteins in whole genomes, we downloaded the protein sequences from the EnsEMBL web site (www.ensembl.org, Hubbard *et al*., 2005). We considered the releases NCBI 35 for *Homo sapiens*, NCBI m34 for *Mus musculus*, WS 140 for *Caenorhabditis elegans* and SGD 1 for *Saccharomyces cerevisiae*. The realease TAIR6 of the *Arabidopsis thaliana* has been downloaded from the TAIR web site (http://www.arabidopsis.org/, Rhee *et al*., 2003).

Two more data sets containing experimental annotated data are used: a data set of 2618 globular proteins deriving from the Yeast GFP fusion databases localized in cytoplasm, nucleus or mitochondria (http://yeastgfp.ucsf.edu, Huh *et al*., 2003) and a set of 499 globular proteins localized in the *Arabidopsis thaliana* chloroplast downloaded from the Plastid Protein Database (http://www.plprot.ethz.ch, Kleffmann *et al*., 2004)

2.2 BaCelLo architecture

Support Vector Machines (SVM) were first introduced by Cortes and Vapnik (1995) and are now broadly used in protein classification tasks. SVMs are able to discriminate two classes of examples by creating a hyperplane that optimally separates them with the best possible margin. Typically the hyperplane is built in a h-dimensional space H in which the examples are mapped by means of feature vectors, that result from the input vectors upon a transformation induced by a kernel function. We used the SVM-light package, version 6, freely available at http://svmlight.joachims.org. We adopted the Radial Basis Function (RBF) kernel since it gives the best performances (data not shown). All the parameters were set as default, except for Gamma and C, which were varied to get the best results.

Our predictor is composed of four support vector machines (SVM) arranged in a decision tree. Each node of the tree is a binary SVM. Different tree architectures were implemented and the most efficient were chosen. The architectures of the trees are shown in Figure 1 and are the same for animals

Table 1a. Number of proteins in the three kingdom specific datasets derived from SWISS-PROT 48

	Plants	Animals	Fungi
Nucleus	121	1166	711
Cytoplasm	58	439	211
Extracellular	41	804	88
Mitochondria	67	188	188
Chloroplast	204		
Total	491	2597	1198

Table 1b. Number of proteins in the training sets extracted from SWISS-PROT 41 and in the testing sets extracted from subsequent releases up to SWISS-PROT 48

	Animals		Fungi	
	Train	Test	Train	Test
Nucleus	803	363	589	122
Cytoplasm	302	137	181	30
Extracellular	632	172	72	16
Mitochondria	153	35	177	11
Total	1890	707	1039	179

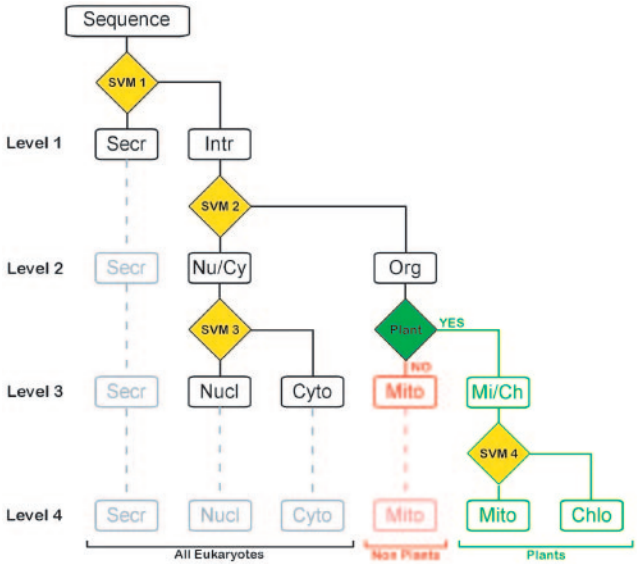


Fig. 1. Architecture of the BaCelLo's decision tree. Abbreviations: Secr: Secretory Pathway, Intr: Intracellular, Org: Organelles, Nu and Nucl: Nucleus, Cy and Cyto: Cytoplasm, Mi and Mito: Mitochondion, Ch and Chlo: Chloroplast.

and fungi, while the plant tree contains an additional node for separating chloroplastic and mitochondrial proteins. Different levels of prediction can be distinguished, each discriminating a different number of classes. For sake of comparison with the other methods we will always consider the level 2, discriminating three classes (Secretory pathway, Nuclear/Cytoplasmic and Organellar) and the level 4, discriminating four and five classes for non plants and plants, respectively.

Table 2. Input vector definition and RBF kernel parameters for the SVMs

SVM	Whole protein frequency	N-ter frequency	C-ter frequency	γ	C	Input vector length
1	+	+	–	3	6	160
2	+	+	+	3	6	280
3	+	–	–	150	2	40
4	+	–	–	150	2	40

SVMs are numbered as in Figure 1.

We used different information as input for SVMs at each node of the tree, depending on the type of discrimination to be performed. Depending on the SVM, we considered the compositions of the whole sequence, of the N-terminal and C-terminal portions. In all cases, both the sequence composition and the sequence profile composition were taken into account. Sequence profiles were obtained aligning with BLAST each sequence with the eukaryotic sequences released in the version 48 of SWISS-PROT. A threshold for the E-value equal to 10^{-4} was used. From the alignment a sequence profile is derived by counting the frequency of each residue in the aligned sequences in each position of the query sequence. The sequence profile composition for a given portion of the protein is obtained by summing up, over all the considered positions, the contributions of each one of the 20 residues. This procedure gives a 20-valued vector that is then normalized. Summing up, three different types of information were considered:

(1) the whole sequence composition, encoded with a 40-valued vector containing both the raw sequence composition (20 components) and the sequence profile composition (20 components);

(2) the N-terminus composition, encoded with a 120-valued vector, containing both the sequence and the profile compositions for three N-terminal portions, formed by the first 20, 40 and 60 residues, respectively;

(3) the C-terminus composition, encoded with a 120-valued vector, containing both the sequence and the profile compositions for three C-terminal portions, formed by the last 20, 50 and 100 residues, respectively;

Different input codes (including a thorough space search for the best input window lengths) have been tried for each node and Table 2 reports the best performing ones, together with the optimal SVM parameters Gamma and C that were selected.

2.3 Evaluation of the performances

We used different accuracy indexes that were computed starting from the confusion matrix $\boldsymbol{Z}$ in which any element z_{ij}, counts the number of examples belonging to the class i and predicted in the class j. First of all, for each class we computed the coverage (Cov) that is the percent of correctly predicted proteins over the total number of proteins belonging to the class. Defining the number of proteins of i^{th} class:

$$x_i = \sum_j z_{ij} \tag{1}$$

coverage can be computed as:

$$Cov(i) = \frac{z_{ii}}{x_i} \cdot 100 \tag{2}$$

The other standard index for the evaluation is the accuracy (*Acc*) that measures the probability of correct prediction for a class:

$$Acc(i) = \frac{z_{ii}}{\sum_j z_{ji}} \cdot 100 \quad (3)$$

When evaluated on very unbalanced databases, the accuracy tends to be low for those classes containing a small number of sequences, since even a very low error rate in such a class can lead to a great number of false positive in the big complementary classes, increasing the denominator in Eq. 3. For these reasons we introduced the Normalized Accuracy (*nAcc*), in which any term of Eq. 3 is divided by the abundance of the respective class in the data set:

$$nAcc(i) = \frac{z_{ii}/x_i}{\sum_j z_{ji}/x_j} \cdot 100 \quad (4)$$

To define a global predictive performance for each class, we used the geometric average (*GAv*) between coverage and normalized accuracy:

$$GAv(i) = \sqrt{Cov(i) \cdot nAcc(i)} \quad (5)$$

In order to evaluate the global performance on all the classes, different parameters are adopted.

Routinely the overall accuracy is computed, defined as the number of correct predictions over the total number of proteins:

$$Q = \frac{\sum_i z_{ii}}{N} \cdot 100 \quad (6)$$

where N is the total number of proteins.

In unbalanced data sets, this parameter is biased towards the performance of the most abundant classes. We introduced the normalized overall accuracy, that counts the number of correct predictions assuming the equiprobability for each class:

$$nQ = \frac{\sum_i z_{ii}/x_i}{K} \cdot 100 \quad (7)$$

where K is number of the classes.

We also used the Generalized Correlation (GC) as proposed by Baldi *et al.* (2000) to analyze the multiclass accuracy. Defining the number of proteins predicted in the i^{th} class:

$$y_i = \sum_j z_{ji} \quad (8)$$

and the matrix:

$$e_{ij} = \frac{x_i y_j}{N} \quad (9)$$

the generalized correlation is computed as

$$GC = \sqrt{\frac{\sum_{ij} \frac{(z_{ij} - e_{ij})^2}{e_{ij}}}{N(K-1)}} \quad (10)$$

It is worth noticing that the generalized correlation does not make use of explicit normalization and can be considered independent of the normalized overall accuracy as defined in Eq. 7.

2.4 Balancing procedure

For all the three datasets the number of sequences for the different classes is highly uneven (Table 1) and the SVMs at each stage discriminate between two classes that are not equally represented in the training set. In the case of mitochondrial versus nuclear/cytoplasmic compartments, for example, the disproportion of the number of sequences is about 8. Under such condition, SVMs tend to over-predict the most abundant class and this can seriously affect the prediction of the under-represented classes (Wang *et al.*, 2004). To solve this problem we adopted the following procedure. For each kingdom, the data set was split in ten subsets. Eight sets are used to train the binary SVM classificators. Each one of the SVMs finds a (h-1)-plane in the h-dimensional hyperspace H of the features; for any point in the feature space a distance from the separating plane was defined. A conventional sign was computed in order to determine in which side the vector point lies with respect to the plane. Routinely a threshold of this 'signed distance' equal to zero is considered for separating the two classes, but a bias can be added to shift the hyperplane (Cortes and Vapnik, 1995). We adopted this possibility to overcome the problem of unbalanced data. The basic idea is to shift the plane in the direction that favors the classification for the less abundant class. Thus a threshold on the 'signed distance' is evaluated on a validation set. The goal is to minimize the unbalance of the predictive performances between the two classes and this can be obtained searching for the threshold on the signed distance that minimizes:

$$| GAv_{k(+)} - GAv_{k(-)} | \quad (11)$$

where *GAv* is the geometric average of the normalized accuracy and the coverage, as defined in Eq. 5. The threshold that minimizes Eq. 11 typically maximizes also the sum of the two geometric averages and leads to the optimal performance. The final performance is then evaluated on the remaining set, called the test set, that is not used to set the SVM parameter nor to pick the optimal threshold. The procedure is repeated ten times, in order to predict each one of the split sets with methods whose parameters have been computed using all the other sets.

Table 3. Performances of BLAST assignment in the RH dataset

Classes	no E-value threshold				E-value < 10^{-3}			
	Cov	*Acc*	*nAcc*	*GAv*	*Cov*	*Acc*	*nAcc*	*GAv*
Nucleus	93.3	98.0	82.4	87.7	98.5	91.9	95.4	96.9
Cytoplasm	86.1	90.1	74.7	80.2	93.2	82.5	81.6	87.2
Secretory Pathway	91.0	99.0	95.0	93.0	97.3	89.7	99.6	98.4
Mitochondria	69.2	90.6	91.4	79.5	81.5	82.2	96.0	88.5
Q	87.8				94.8			
nQ	84.9				92.6			
GC	0.81				0.92			
Not Assigned (%)	0.0				14.1			

Not assigned proteins are sequences for which no homologous (under the E-value threshold) can be found in the RH dataset. Values are normalized on the number of assigned sequences. Abbreviations: Cov: Coverage, Acc: Accuracy, nAcc: normalized Accuracy, GAv: Geometric Average, Q: number of proteines correctly predicted, nQ: normalized number of proteins correctly predicted, GC: Generalized Correlation: see the Materials and Methods section for their definition.

3 RESULTS AND DISCUSSION

3.1 Necessity of a non-redundant training set

Many available predictors with the exception of LocTree (Nair and Rost, 2005) were implemented using redundant training sets. Sequences sharing up to 95% identity were routinely selected. The rationale for this is the fact that little differences in the sequences can lead to different subcellular location. It is therefore important to quantify to which extent sequence identity affects subcellular location and to which extent redundancy in the training set leads to a tool with poor generalization capability when predicting sequences scarcely related to those considered for training. The most largely adopted of these data sets was firstly released by Reinhardt and Hubbard (1998) (RH Dataset) and contains 2427 eukaryotic proteins divided into four subcellular classes: extracellular (325 sequences), cytoplasm (684), nucleus (1097) and mitochondria (321).

Table 4. 10-fold crossvalidation performances of BaCelLo on three kingdom-specific datasets

Level	Classes	Plants							Animals							Fungi						
		Cov	*nAcc*	*Acc*	*GAv*	*nQ*	*Q*	*GC*	*Cov*	*nAcc*	*Acc*	*GAv*	*nQ*	*Q*	*GC*	*Cov*	*nAcc*	*Acc*	*GAv*	*nQ*	*Q*	*GC*
1	Secr	85.4	95.3	64.8	90.2	90.6	94.9	0.72	90.8	95.6	90.7	93.2	93.3	94.3	0.87	94.3	97.7	76.9	96.0	96.0	97.5	0.84
	Intr	95.8	86.7	98.6	91.1				95.8	91.2	95.8	93.5				97.7	94.5	99.5	96.1			
2	Nucl/Cyto	80.4	76.0	80.4	78.2	84.4	84.7	0.73	92.9	82.9	94.6	87.8	86.6	91.0	0.78	91.2	83.2	96.7	87.1	89.0	89.9	0.78
	Secr	85.4	89.8	64.8	87.6				90.8	85.0	90.7	87.9				94.3	92.8	76.9	93.5			
	Mito/Chlo	87.5	88.2	91.9	87.8				76.1	93.6	66.2	84.4				81.4	91.8	69.5	86.4			
3	Nucl	71.9	69.1	75.7	70.5	74.1	79.2	0.65	64.8	67.8	84.9	66.3	74.2	73.8	0.67	67.1	65.7	87.0	66.4	75.8	70.1	0.66
	Cyto	51.7	65.5	46.9	58.2				65.3	60.3	41.4	62.8				60.2	62.3	39.4	61.2			
	Secr	85.4	81.3	64.8	83.3				90.8	83.1	90.7	86.9				94.3	90.6	76.9	92.4			
	Mito/Chlo	87.5	78.1	91.9	82.7				76.1	87.4	66.2	81.6				81.4	83.8	69.5	82.6			
4	Nucl	71.9	66.8	75.7	69.3	66.6	68.2	0.59														
	Cyto	51.7	61.6	46.9	56.4																	
	Secr	85.4	80.0	64.8	82.7																	
	Mito	50.7	77.3	54.0	62.6																	
	Chlo	73.0	53.6	76.4	62.6																	

Abbreviations: See caption of Table 3 Secr: Secretory Pathway, Intr: Intracellular, Nucl: Nucleus, Cyto: Cytoplasm, Mito: Mitochondria, Chlo: Chloroplast.

We predicted the localization for every protein in the RH dataset with a BLAST search on the same dataset. The results of the assignment based on the closest non identical homologue are shown on Table 3. Setting a threshold for the E-value equal to 10^{-3} (corresponding approximately to a local sequence identity higher than 25%) 86% of the RH sequences are similar to at least another sequence of the set. A simple procedure based on transfer annotation is then able to correctly assign 94% of the proteins (1974 chains). When no E-value threshold is considered, and the annotation is transferred from the closest homologous regardless of the sequence identity level, the accuracy is still as high as 88%. Notably, this performance is similar to that achieved by methods trained on the same RH dataset: LOCSVMpsi (Xie *et al*., 2005), ESLpred (Bhasin and Raghava, 2005) and SubLoc (Hua and Sun, 2001) that reach overall accuracies (*Q*) as high as 90%, 88% and 79%, respectively.

The goal of a good predictor is to assign a subcellular localization especially when no homology is detectable and for this a non-redundant data set needs to be selected.

3.2 BaCelLo performances

Three non-redundant sequence sets were selected, for animals, fungi and plants, respectively. In each set the sequences share less than 30% identity. We generated three eukaryotic datasets in order to take into account the differences in the subcellular localization mechanism between evolutionary distant kingdoms.

BaCelLo is a system of SVMs organized in a tree structure, which exploits the information from the sequence composition and from the sequence profile composition. The input of the SVMs considers the whole sequence and different portion of the N- and C- termini, which are likely to contain localization signals. However it does not make explicit search for localization signal or implement annotation transfer by homology. Different tree architectures have been tried and the best performing one is adopted. The performance of BaCelLo, computed on the test sets with a rigorous 10-fold cross validation procedure, is shown in Table 4. The decision tree structure of our system allows to predict the subcellular location at different stages with different accuracy. In all the kingdoms extracellular/secretory proteins are well discriminated from intracellular proteins. When the normalized overall accuracy is considered (*nQ*), BaCelLo at the first level of the tree (Fig. 1) correctly discriminates 96%, 93% and 91% of the proteins from fungi, animals and plants, respectively. This level of discrimination is achieved using information from the whole sequence and from the N-terminal portions, where signal peptides are supposed to be. Adding one more level, BaCelLo discriminates among 3 classes: extracellular/secretory, nuclear/cytoplasmic and mitochondrial/chloroplastic. The normalized overall accuracy ranges from 89% in fungi to 84% in plants. At this stage the prediction exploits the information extracted from the whole sequence and from both the N- and the C-terminal portions. In the next step of the decisional tree nuclear and cytoplasmic proteins are discriminated. Since up to date no conserved nor general localization signal is known, this step is done using only information about the whole protein. The performance on the four classes ranges between 75% (fungi) and 74% (plants). For plant proteins an additional step is introduced to separate mitochondrial from chloroplastic proteins. Trying different input coding we verified that there is no advantage in using information from both termini portions (data not shown). For this reason the step exploits only the information from the whole protein sequence. The overall accuracy achieved for plant proteins is then 66% on 5 classes.

The data in Table 4 show that the best performance is reached in discriminating extracellular proteins. Prediction at level 2, where proteins of organelles are discriminated, is very good, while the distinction between nuclear and cytoplasmic proteins (level 3) and between chloroplastic and mitochondrial proteins (level 4) is more problematic, leading to a quite poor coverage for cytoplasmic and mitochondrial proteins and a quite poor accuracy for chloroplastic proteins.

It is evident that the accuracy value of each class is strongly influenced by the dimension of the class, being remarkably higher in most abundant classes, namely nuclear in non-plants or nuclear

Table 5. Comparative performances on the test set

3 classes		Fungi											Animals										
Method		**a**	b	c	d	e	f	g	h	i	j	k	**a**	b	c	d	e	f	g	h	i	j	k
Nucl/Cyto	*Cov*	**88.8**	87.5	88.1	86.8	97.4	98.0	94.1	96.1	87.5	90.8	*85.5*	**93.2**	88.8	92.6	84.0	92.6	95.2	96.6	97.6	89.4	90.6	*88.6*
	nAcc	**93.4**	75.9	56.6	54.0	48.1	63.8	55.3	59.9	65.6	76.5	*90.4*	**71.7**	67.9	60.7	47.5	44.4	54.1	60.7	62.9	70.8	66.6	*87.5*
	GAv	**91.1**	81.5	70.6	68.5	68.4	79.1	72.1	75.9	75.8	83.3	*87.9*	**81.7**	77.6	75.0	63.2	64.1	71.8	76.6	78.4	79.6	77.7	*88.0*
Secr	*Cov*	**93.8**	81.3	56.3	50.0	31.3	62.5	87.5	81.3	87.5	81.3	*43.8*	**85.5**	84.9	56.4	47.7	20.3	53.5	76.2	84.9	85.5	82.0	*62.8*
	nAcc	**99.3**	69.4	100	93.8	96.0	100	96.4	96.9	95.7	95.4	*100*	**90.7**	85.8	93.9	80.6	83.4	92.4	94.2	95.9	94.6	91.4	*100*
	GAv	**96.5**	75.1	75.0	68.5	54.8	79.1	91.8	88.8	91.5	88.1	*66.2*	**88.1**	85.3	72.8	62.0	41.1	70.3	84.7	90.2	89.9	86.6	*79.2*
Mito	*Cov*	**100**	63.6	63.6	63.6	63.6	81.8	36.4	54.5	66.7	90.9	*81.8*	**68.6**	60.0	54.3	48.6	54.3	58.3	57.1	54.5	74.3	65.7	*71.4*
	nAcc	**90.5**	94.2	94.1	74.0	98.0	97.6	93.3	97.6	88.6	94.5	*95.4*	**90.4**	85.1	84.7	75.8	81.2	88.1	95.4	96.8	89.0	88.5	*91.0*
	GAv	**95.1**	77.4	77.4	68.6	78.9	89.4	58.3	72.9	76.9	92.7	*88.3*	**78.7**	71.5	67.8	60.7	66.4	71.7	73.8	72.6	81.3	76.3	*80.6*
nQ		**94.2**	77.5	69.3	66.8	64.1	80.8	77.7	77.3	80.6	87.6	*70.4*	**82.4**	77.9	67.8	60.1	55.7	69.0	76.6	79.0	83.0	79.4	*74.3*
GC		**0.79**	0.57	0.53	0.45	0.58	0.77	0.62	0.70	0.62	0.70	*0.59*	**0.71**	0.64	0.48	0.38	0.36	0.57	0.70	0.75	0.70	0.67	*0.60*
Assigned to other locations		**-**	-	7.9	-		-	-	-	-	-	*14.5*	**-**	-	4.1	-	-	-	-	-	-	-	*14.0*
4 classes		Fungi											Animals										
Nucl	*Cov*	**66.4**	66.4	71.1	70.5	84.4	88.5					*62.3*	**66.1**	62.2	70.2	67.8	79.1	80.2					*73.3*
	nAcc	**71.3**	66.9	44.2	38.4	37.5	51.0					*63.5*	**56.4**	49.5	43.0	37.2	35.8	38.7					*64.2*
	GAv	**68.8**	66.6	56.1	52.0	56.3	67.2					*62.9*	**61.1**	55.5	54.9	50.2	53.2	55.7					*68.6*
Cyto	*Cov*	**56.7**	46.7	36.7	23.3	23.3	30.0					*56.7*	**54.0**	38.2	40.9	21.9	28.5	29.2					*46.0*
	nAcc	**65.4**	50.3	46.2	32.7	35.0	40.1					*70.4*	**50.7**	42.0	48.5	28.6	36.1	44.9					*63.1*
	GAv	**60.9**	48.5	41.2	27.6	28.6	34.7					*63.2*	**52.3**	40.1	44.5	25.0	32.1	36.2					*53.9*
Secr	*Cov*	**93.8**	81.3	56.3	50.0	31.3	62.5					*43.8*	**85.5**	84.9	56.4	47.7	20.3	53.5					*62.8*
	nAcc	**99.1**	61.7	100	92.4	82.4	100					*100*	**88.4**	80.0	93.4	67.1	76.3	90.1					*100*
	GAv	**96.4**	70.8	75.0	68.0	50.8	79.1					*66.2*	**86.9**	82.4	72.6	56.6	39.4	69.4					*79.2*
Mito	*Cov*	**100**	63.6	63.6	63.6	63.6	81.8					*81.8*	**68.6**	60.0	54.3	48.6	54.3	58.3					*71.4*
	nAcc	**79.6**	83.6	86.5	70.0	90.5	91.6					*89.2*	**86.2**	76.8	80.5	69.3	73.6	85.4					*86.7*
	GAv	**89.2**	72.9	74.2	66.7	75.9	86.6					*85.4*	**76.9**	67.9	66.1	58.0	63.2	70.6					*78.7*
nQ		**79.2**	64.3	56.9	51.9	50.7	65.7					*61.1*	**68.5**	61.3	55.5	46.5	45.5	55.3					*63.4*
GC		**0.68**	0.50	0.47	0.39	0.49	0.65					*0.54*	**0.60**	0.53	0.43	0.32	0.31	0.48					*0.54*
Assigned to other locations		**-**	-	7.9	-	-	-					*14.5*	**-**	-	4.1	-	-	-					*14.0*

a: BaCelLo, **b:** Loctree (Nair and Rost, 2005), **c**: Psort II (Nakai and Horton, 1999), **d**: SubLoc (Hua and Sun, 2001), **e**: ESLpred (Bhasin and Raghava, 2004), **f**: LOCSVMpsi (Xie *et al*., 2005), **g**: SLP-local (Matsuda *et al*., 2005), **h**: Protein Prowler (Boden and Hawkins, 2005), **i**: TARGETp (Emanuelsson *et al*., 2000), **j**: PredoTar (Small *et al*., 2004) , **k**: pTARGET (Guda and Subramaniam, 2005).
pTARGET results are highlighted in italic since it used for training proteins released after the release 41 of SWISS-PROT; Abbreviations: see caption for Table 3.

Table 6. Prediction of protein localization for whole genomes

	H.sapiens	*M.musculus*	*C.elegans*	*S.cerevisiae*	*A.thaliana*
Nucl	8725 *(26%)*	6811 *(24%)*	5878 *(23%)*	2078 *(32%)*	7050 *(25%)*
Cyto	10399 *(31%)*	10909 *(31%)*	6674 *(26%)*	1611 *(25%)*	6033 *(20%)*
Secr	4960 *(15%)*	5417 *(15%)*	4767 *(19%)*	227 *(3%)*	3001 *(10%)*
Mito	2452 *(7%)*	2793 *(8%)*	1516 *(6%)*	971 *(15%)*	963 *(3%)*
Chlo					4875 *(16%)*
Memb	7017 *(21%)*	7610 *(22%)*	6879 *(25%)*	1657 *(25%)*	8078 *(26%)*
Total	33553	35340	25714	6544	30600

For each species the number of proteins predicted in each localization and the Percentage with respect to the total are shown. Abbreviations: Secr: Secretory Pathway, Nucl: Nucleus, Cyto: Cytoplasm, Mito: Mitochondria, Chlo: Chloroplast, Memb: Membrane.

and chloroplastic in plants. This is due to the fact that also a great rate of false positives on the most abundant classes gives a low number of false positive on the other classes and then scarcely affects the accuracy. Since the proportion of the classes in the data set does not reflect any reliable a priori hypothesis, a more meaningful evaluation can be carried out considering the normalized parameters, defined in the Material and Methods section. It is worth noticing that the balancing procedure leads to performances in which the coverage and the normalized accuracy are similar for each class, except in the case of chloroplasts and mitochondria in plants, where the latter tend to be under-predicted. This may be due to the under representation of mitochondrial proteins from plants (67 examples of non-redundant mitochondrial proteins are known in plants while 188 are known in both animals and fungi).

The performances of the three kingdom-specific predictors are quite similar, but it is worth to noticing that merging the fungi and the animal proteins, similarly to what the other predictors do, leads to a poorer performance, in particular for fungi (data not shown).

3.3 Comparison with the other methods

The performance of BaCelLo has been compared to those of the best publicly available methods for the prediction of the subcellular localization. Some of them discriminate among three classes, namely TARGETp (Emanuelsson *et al*., 2000), ProteinProwler (Boden and Hawkins, 2005), SLP-local (Matsuda *et al*., 2005) and Predotar (Small *et al*., 2004); others discriminate among four classes in animals and five classes in plants and are Loctree, SubLoc, ESLpred and LOCSVMpsi. pTARGET and Psort II (Nakai and Horton, 1999) discriminate more classes than BaCelLo does, however we did not consider these classes since very few redundant examples are known. All the considered predictors, but pTARGET, have been trained on a dataset at most derived from SWISS-PROT, rel. 41. Then, in order to compare the performances, we retrained BacCelLo using only the subset of training sequences that were yet included in the release 41 of SWISS-PROT. The test and the comparison has been performed on the remaining sequences, up to release 48, that, by construction, are less than 30% identical to those of the training. It is important to note that some 30% proteins of this test set share identity with proteins included in SWISS-PROT 41, so that methods different from BaCelLo can still have some homology with the training set. Moreover, when comparing with pTARGET it has to be kept in mind that it was developed using proteins derived from release 46 of SWISS-PROT and then that it has been successively updated.

Predictions were run with default options, except for LOCSVMpsi, for which the four class classification option was selected. For the predictors that consider more than four classes, proteins predicted in classes not considered by BaCelLo are considered as badly assigned.

On the fungi dataset (Table 5), BaCelLo outperforms the other methods by at least 7% in terms of normalized overall accuracy, for the three classes predictions and 14% for the four classes ones. It performs remarkably better in discriminating secreted and mitochondrial proteins, even when compared with TARGETp, Proteins Prowler and Predotar that are explicitly designed to recognize signal and target peptides for these localizations. Furthermore we achieve the best average prediction (*GAv*) in each class and, outperforming other methods, a good balancing between coverage (*Cov*) and normalized accuracy (*nAcc*). As a general consideration, most of the methods achieve on this test a worse performance than that reported in the original papers, corroborating the notion that the redundancy of their data sets affects the generalization performances. Similar considerations are valid for the animal test set (Table 5). In this case the improvement with respect to other methods is about 5% on four classes. All the predictors perform worse on the animal set than on the fungi set. However the large improvement of the BaCelLo performances in predicting fungal proteins confirms the advantage in implementing a fungal-specific predictor.

3.4 Genome predictions

We adopted BaCelLo for high-throughput prediction of protein subcellular localization. Five proteomes have been tested in order to estimate the composition of protein localization. We predicted the localization of proteins from *H. sapiens*, *M. musculus* and *C. elegans* for animals, *S. cerevisiae* for fungi and *A. thaliana* for plants. Membrane proteins predicted with SpepLip (Fariselli *et al*., 2003) and ENSEMBLE (Martelli *et al*., 2003) were excluded from the set predicted with BaCelLo. The results of this large scale analysis are reported in Table 6, where both the number and the frequency of proteins predicted in each class are listed. In all the examined proteomes the sum of nuclear and cytoplasmic proteins accounts for about 50-60% of all proteins. Protein localization composition for Human and Mouse are very similar, as

expected, while *C. elegans* contains about the same number of secreted and membrane proteins as the other animals, but significantly fewer proteins in the nucleus and cytoplasm. *S. cerevisiae* is a unicellular organism, not endowed with an extracellular matrix nor communicating with other cells. Interestingly and accordingly, only 3% of its proteome is predicted as secreted. Concerning plants, up to 19% of the *A. thaliana* proteins are directed to organelles and more than 80% of them are directed to chloroplasts. The results are available at http://www.biocomp.unibo.it/bacello.

3.5 Comparison with high-throughput experimental data

As a proof of the reliability of the prediction of BaCelLo we compared predictions with data obtained with high-throughput methods in *A.thaliana* and in yeast. The PLPROT data base contains 499 annotated chloroplast proteins from *A.thaliana* and we correctly assign 53% of them, an amount similar to that reported for LocTree. From the Yeast GFP fusion database we extracted 2618 sequences experimentally annotated: 483 'mitochondrial', 496 'nuclear', 818 'cytoplasmic' and 821 'nuclear and cytoplasmic'. The performance on mitochondrial proteins is 87%. The rate of correct prediction for nuclear and cytoplasmic proteins reaches about 50%. Nevertheless considering 'nuclear', 'cytoplasmic' and 'nuclear and cytoplasmic' proteins together, level 2 of our predictor correctly assigns 88% of proteins.

4 CONCLUSIONS

BaCelLo is a new method for predicting the subcellular localization of a protein sequence from animals, fungi or plants. Three kingdom-specific sets of parameters have been inferred from non-redundant data sets of annotated proteins. This is at the basis of the implementation of the first *de novo* predictor specific for animals and fungi. The reduction of the redundancy of the training sets guarantees the generalization capability of BaCelLo. We prove that predictors trained on very redundant data sets don't perform better than a simple annotation transfer based on a BLAST search.

The key feature of BaCelLo is the procedure to balance the prediction scoring indexes, overcoming the biases in the dataset composition. BaCelLo outperforms other predictors on a test set of proteins that don't share identity with sequences used for training. Furthermore BaCelLo can be easily used for large scale analysis of whole genomes to produce an estimate and annotation of the protein content in each subcellular compartment.

BaCelLo is available at the site http://www.biocomp.unibo.it/bacello.

ACKNOWLEDGEMENTS

RC acknowledges the receipt of the following grants: PNR 2001-2003 (FIRB art.8) for a project on Bioinformatics for Genomics and Proteomics, a FIRB 2003 LIBI—International Laboratory of Bioinformatics and the support to the Bologna node of the Biosapiens Network of Excellence project within the European Union's VI Framework Programme (contract number LSHG-CT-2003-503265). PF acknowledges MIUR for a grant on Proteases. AP and PLM are supported by a FIRB 2003-LIBI grant.

REFERENCES

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Bairoch,A., Apweiler,R., Wu,C.H., Barker,W.C., Boeckmann,B., Ferro,S., Gasteiger,E., Huang,H., Lopez,R., Magrane,M., Martin,M.J., Natale,D.A., O'Donovan,C., Redaschi,N. and Yeh,L.S. (2005) The Universal Protein Resource (UniProt). *Nucleic Acids Res.*, **33**, D154–D159.

Baldi,P., Brunak,S., Chauvin,Y., Andersen,C.A.F. and Nielsen,H. (2000) Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, **16**, 412–424.

Bendtsen,J.D., Jensen,L.J., Blom,N., Von Heijne,G. and Brunak,S. (2004) Feature-based prediction of non-classical and leaderless protein secretion. *Protein Eng. Des. Sel.*, **17**, 349–356.

Bhasin,M. and Raghava,G.P. (2004) ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. *Nucleic Acids Res.*, **32**, W414–419.

Boden,M. and Hawkins,J. (2005) Prediction of subcellular localization using sequence-biased recurrent networks. *Bioinformatics*, **21**, 2279–2286.

Cortes,C. and Vapnik,V. (1995) Support vector networks. *Mach. Learn.*, **20**, 273–293.

Emanuelsson,O., Nielsen,H., Brunak,S. and von Heijne,G. (2000) Predicting subcellular localization of proteins based on their N-terminal residue sequence. *J. Mol. Biol.*, **300**, 1005–1016.

Fariselli,P., Finocchiaro,G. and Casadio,R. (2003) SPEPlip: the detection of signal peptide and lipoprotein cleavage sites. *Bioinformatics*, **19**, 2498–2499.

Fried,H. and Kutay,U. (2003) Nucleocytoplasmic transport: taking an inventory. *Cell. Mol. Life Sci.*, **60**, 1659–1688.

Gonsalvez,G.B., Urbinati,C.R. and Long,R.M. (2005) RNA localization in yeast: moving towards a mechanism. *Biol. Cell.*, **97**, 75–86.

Guda,C. and Subramaniam,S. (2005) pTARGET a new method for predicting protein subcellular localization in eukaryotes. *Bioinformatics*, **21**, 3963–3969.

Hua,S. and Sun,Z. (2001) Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, **17**, 721–728.

Hubbard,T., Andrews,D., Caccamo,M., Cameron,G., Chen,Y., Clamp,M., Clarke,L., Coates,G., Cox,T., Cunningham,F., Curwen,V., Cutts,T., Down,T., Durbin,R., Fernandez-Suarez,X.M., Gilbert,J., Hammond,M., Herrero,J., Hotz,H., Howe,K., Iyer,V., Jekosch,K., Kahari,A., Kasprzyk,A., Keefe,D., Keenan,S., Kokocinsci,F., London,D., Longden,I., McVicker,G., Melsopp,C., Meidl,P., Potter,S., Proctor,G., Rae,M., Rios,D., Schuster,M., Searle,S., Severin,J., Slater,G., Smedley,D., Smith,J., Spooner,W., Stabenau,A., Stalker,J., Storey,R., Trevanion,S., Ureta-Vidal,A., Vogel,J., White,S., Woodwark,C. and Birney,E. Ensembl (2005). *Nucleic Acids Res.*, **33**, D447–D453.

Huh,W.K., Falvo,J.V., Gerke,L.C., Carroll,A.S., Howson,R.W., Weissman,J.S. and O'Shea,E.K. (2003) Global analysis of protein localization in budding yeast. *Nature*, **425**, 686–91.

Izeta,A., Malcomber,S., O'Rourke,D., Hodgkin,J. and O'Hare,P. (2003) A C-terminal targeting signal controls differential compartmentalisation of Caenorhabditis elegans host cell factor (HCF) to the nucleus or mitochondria. *Eur. J. Cell. Biol.*, **82**, 495–504.

Kleffmann,T., Russenberger,D., von Zychlinski,A., Christopher,W., Sjolander,K., Gruissem,W. and Baginsky,S. (2004) The Arabidopsis thaliana chloroplast proteome reveals pathway abundance and novel protein functions. *Curr. Biol.*, **14**, 354–362.

Lee,C.M., Sedman,J., Neupert,W. and Stuart,R.A. (1999) The DNA helicase, Hmi1p, is transported into mitochondria by a C-terminal cleavable targeting signal. *J. Biol. Chem.*, **274**, 20937–20942.

Marcotte,E.M., Xenarios,I., van Der Bliek,A.M. and Eisenberg,D. (2000) Localizing proteins in the cell from their phylogenetic profiles. *Proc. Natl. Acad. Sci. USA*, **97**, 12115–12120.

Martelli,P.L., Fariselli,P. and Casadio,R. (2003) An ENSEMBLE machine learning approach for the prediction of all-alpha membrane proteins. *Bioinformatics*, **19**, i205–i211.

Matsuda,S., Vert,J.P., Saigo,H., Ueda,N., Toh,H. and Akutsu,T. (2005) A novel representation of protein sequences for prediction of subcellular location using support vector machines. *Protein Science*, **14**, 2804–2813.

Nair,R. and Rost,B. (2005) Mimicking cellular sorting improves prediction of subcellular localization. *J. Mol. Biol.*, **348**, 85–100.

Nakai,K. and Horton,P. (1999) PSORT: a program for detecting the sorting signals of proteins and predicting their subcellular localization. *Trends Biochem. Sci.*, **24**, 34–35.

Nickel,W. (2003) The mystery of nonclassical protein secretion. A current view on cargo proteins and potential export routes. *Eur. J. Biochem.*, **270**, 2109–2119.

Nielsen,H., Engelbrecht,J., Brunak,S. and von Heijne,G. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Eng*., **10**, 1–6.

Reinhardt,A. and Hubbard,T. (1998) Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Res*., **26**, 2230–2236.

Rhee,S.Y., Beavis,W., Berardini,T.Z., Chen,G., Dixon,D., Doyle,A., Garcia-Hernandez,M., Huala,E., Lander,G., Montoya,M., Miller,N., Mueller,L.A., Mundodi,S., Reiser,L., Tacklind,J., Weems,D.C., Wu,Y., Xu,I., Yoo,D., Yoon,J. and Zhang,P. (2003) The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. *Nucleic Acids Res*., **31**, 224–228.

Scott,M.S., Thomas,D.Y. and Hallett,M.T. (2004) Predicting subcellular localization via protein motif co-occurrence. *Genome Res*., **14**, 1957–1966.

Small,I., Peeters,N., Legeai,F. and Lurin,C. (2004) Predotar: A tool for rapidly screening proteomes for N-terminal targeting sequences. *Proteomics*, **6**, 1581–1590.

Wang,M., Yang,J., Liu,G.P., Xu,Z.J. and Chou,K.C. (2004) Weighted-support vector machines for predicting membrane protein types based on pseudo-residue composition. *Protein Eng. Des. Sel*., **17**, 509–516.

Xie,D., Li,A., Wang,M., Fan,Z. and Feng,H. (2005) LOCSVMPSI: A web server for subcellular localization of eukaryotic proteins. *Nucleic Acids Res*., **33**, W105–W110.

Yamada,H., Chounan,R., Higashi,Y., Kurihara,N. and Kido,H. (2004) Mitochondrial targeting sequence of the influenza A virus PB1-F2 protein and its function in mitochondria. *EBS Lett*., **578**, 331–336.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e417–e423
doi:10.1093/bioinformatics/btl256

Semi-supervised analysis of gene expression profiles for lineage-specific development in the *Caenorhabditis elegans* embryo

Yuan Qi[1], Patrycja E. Missiuro[2], Ashish Kapoor[3], Craig P. Hunter[4], Tommi S. Jaakkola[1], David K. Gifford[1,*] and Hui Ge[2,*]

[1]Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139, USA, [2]Whitehead Institute, 9 Cambridge Center, Cambridge, MA 02142, USA, [3]Microsoft Research, 1 Microsoft Way, Redmond, WA 98052, USA and [4]Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA 02138, USA

ABSTRACT

Motivation: Gene expression profiling is a powerful approach to identify genes that may be involved in a specific biological process on a global scale. For example, gene expression profiling of mutant animals that lack or contain an excess of certain cell types is a common way to identify genes that are important for the development and maintenance of given cell types. However, it is difficult for traditional computational methods, including unsupervised and supervised learning methods, to detect relevant genes from a large collection of expression profiles with high sensitivity and specificity. Unsupervised methods group similar gene expressions together while ignoring important prior biological knowledge. Supervised methods utilize training data from prior biological knowledge to classify gene expression. However, for many biological problems, little prior knowledge is available, which limits the prediction performance of most supervised methods.

Results: We present a Bayesian semi-supervised learning method, called BGEN, that improves upon supervised and unsupervised methods by both capturing relevant expression profiles and using prior biological knowledge from literature and experimental validation. Unlike currently available semi-supervised learning methods, this new method trains a kernel classifier based on labeled and unlabeled gene expression examples. The semi-supervised trained classifier can then be used to efficiently classify the remaining genes in the dataset. Moreover, we model the confidence of microarray probes and probabilistically combine multiple probe predictions into gene predictions. We apply BGEN to identify genes involved in the development of a specific cell lineage in the *C. elegans* embryo, and to further identify the tissues in which these genes are enriched. Compared to K-means clustering and SVM classification, BGEN achieves higher sensitivity and specificity. We confirm certain predictions by biological experiments.

Availability: The results are available at http://www.csail.mit.edu/~alanqi/projects/BGEN.html

Contact: hge@wi.mit.edu or gifford@mit.edu

*To whom correspondence should be addressed.

1 INTRODUCTION

Gene expression profiling is a powerful approach to probe global transcriptional programs underlying biological processes. However, it is a challenge to identify candidate genes with high sensitivity and specificity from large compendia of gene expression profiles. For example, in order to uncover transcriptional changes relevant to the development of certain cell types, gene expression profiles are often compared between wild-type animals and mutants that lack or contain an excess of the cell types (Reinke *et al*., 2000; Furlong *et al*., 2001; Gaudet & Mango, 2002; Robertson *et al*., 2004; Baugh *et al*., 2005). Genes that are spatially or temporally enriched can be identified in this way and then tested to confirm their expression patterns. In these cases, gene expression data are usually obtained from whole animals instead of single cells, so differential expression may be partially masked.

Unsupervised clustering methods have been applied to expression profiles to identify candidate genes (Eisen et al., 1998). Clustering methods group together genes with similar expression profiles by modeling the distribution of an entire dataset. However, they do not incorporate knowledge about genes that are already known to be differentially expressed. Consequently, genes clustered together are coherent in terms of expression profiles, yet they may have diverse biological functions.

Another approach to identify candidate genes is to use supervised classification methods. These methods train a model using prior biological knowledge of gene expression, including known regulators and experimentally confirmed candidate genes, and use the trained model for predictions on other genes. However, for many biological processes, either only a few key regulators have been identified, or only a few candidates are experimentally verified. Most classification methods, including Support Vector Machines (SVMs), use training data on known regulators and confirmed candidate genes. Therefore, with a limited amount of training data, it is difficult for supervised methods to achieve accurate predictions.

We propose a semi-supervised learning method that combines the advantages of supervised classification with the benefits of unsupervised clustering. We call this method *BGEN* (Bayesian

GENeralization from examples). By using information from both prior biological knowledge and the entire expression dataset, BGEN allows us to perform accurate predictions even when we only have scarce information about the known regulators. There have been a large number of approaches proposed in recent years for semi-supervised learning and the spectrum of these approaches include random walks, spectral methods (Belkin & Niyogi, 2004; Joachims, 2003; Zhou *et al*., 2004; Zhu *et al*., 2003), and information-regularization (Szummer & Jaakkola, 2003). BGEN differentiates itself from these previous semi-supervised learning approaches in the following ways. First, it provides a principled kernel classifier to classify new data points. Second, we offer a computationally efficient way to choose parameters of the method. Third, specific to microarray data, BGEN explicitly models probe confidence and probabilistically combines predictions from multiple probes corresponding to the same gene.

We apply BGEN to analyze development and differentiation of a specific cell lineage in the *C. elegans* embryo. *C. elegans* is a free-living soil nematode widely used in developmental biology. The adult nematode contains 959 somatic cells. Embryonic cell divisions from a fertilized egg have been traced by microscopy and the cell division patterns are invariant (Sulston *et al*., 1983). The early asymmetric divisions produce six founder cells: AB, MS, E, C, D and P4. Each of these founder cells maintain a distinct pace of cell divisions and produce a specific subset of tissues and cell types. In this paper, we focus on the differentiation of the C lineage, which mainly gives rise to epidermis and muscle cells.

Using previously published expression profiles of wild-type and mutant *C. elegans* embryos (Baugh *et al*., 2005), we identify genes enriched in C lineage and compare the prediction results of BGEN to those of K-means clustering and SVM classification. BGEN outperforms them with improved sensitivity and specificity. We further classify the candidate C-lineage genes from the whole genome into two sub-categories: epidermis enriched genes and muscle enriched genes. The classification is validated by the experimental results obtained by Baugh *et al*. (2005). To further validate our methodology, we experimentally test one gene predicted to be enriched in C-lineage epidermis cells and one gene predicted to be enriched in C-lineage muscle cells. Our experimental results are consistent with our predictions.

2 APPROACH

We begin with a gene expression compendium, $\mathbf{X} = \{\mathbf{x}_1,..,\mathbf{x}_{n+m}\}$ where $\mathbf{x}_i$ is the feature vector extracted from the gene expression of probe i. We also have a few (n) labeled genes and their corresponding probes, for which $\mathbf{X}_L = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ are labeled as $\mathrm{t}_L=\{t_1,\ldots,t_n\}$, and many unlabeled probes $\mathbf{X}_U = \{\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+m}\}$. Each label t_i is a binary variable. For identification of C-lineage specific genes, labels 1 and -1 correspond to C-lineage and non-C-lineage genes, respectively. For classification among C-lineage candidate genes, labels 1 and -1 correspond to epidermis and muscle enriched genes, respectively.

Similar to traditional classification methods, we will classify a data point x_i based on a classifier w. Given w, the probability of the label $t_i = 1$ for $\mathbf{x}_i$ in $\mathbf{X}$ is

$$p(t_i \mid \mathbf{x}_i, \mathbf{w}) = \Theta(t_i \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}_i)) \tag{1}$$

where $\Theta(\cdot)$ is a link function that maps a continuous unbounded value into a value between 0 and 1, and $\phi(\cdot)$ is a basis function, allowing nonlinear separation of data points. Equation (1) is known as the likelihood function of the data $(t_i, \mathbf{x}_i)$. We assume that the data labels are conditionally independent of each other given the input and the classifier, such that $p(\boldsymbol{t}_L \mid \mathbf{X}_L, \mathbf{w}) = \prod_{i:i\in\{1,2,\ldots,n\}} \Theta(t_i \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}_i))$. Later, we will discuss the likelihood function in more detail.

What distinguishes BGEN from traditional classification or clustering methods is the following: while traditional methods uses either labeled or unlabeled information, BGEN employs the information in both labeled and unlabeled data points. We achieve this by both assigning a data dependent prior $p(\mathbf{w} \mid \mathbf{X})$, which contains the information in *unlabeled* data points $\mathbf{X}_U$, and using the likelihood $p(\mathbf{t}_L \mid \mathbf{X}_L, \mathbf{w})$, which encodes labeled information. We fuse the information in labeled and unlabeled data points by the Bayes rule to compute the posterior distribution $p(\mathbf{w} \mid \mathbf{X}, \boldsymbol{t}_\mathbf{L})$.

Unlike the maximum likelihood or maximum a posteriori approach, which are both point estimates of w for prediction, we average our predictions for t_i based on the posterior distribution $p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}_L)$ to classify unlabeled data points. Note that when given a new data point that is not in the training set $\mathbf{X}$, we can easily classify it based on the classifier posterior $p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}_L)$.

Moreover, in microarray datasets, a gene often corresponds to multiple probes. Therefore, we combine probabilistic predictions of multiple probes to classify their corresponding gene as well as to obtain classification confidence.

In the following subsections we present the prior and the likelihood distributions, describe how to compute the posterior distributions for classifier $\mathbf{w}$ and for label t_i, and show how to combine multiple probe predictions for gene classification, and describe experimental approaches to confirm our predictions.

2.1 From graph regularization to prior on classifiers

The prior plays a significant role in semi-supervised learning, especially when there is only a small amount of labeled data. In those cases, the prior greatly influences the posterior distribution, since the information from the data likelihood is relatively weak.

It is not an easy task to design a sensible prior on $\mathbf{w}$ that incorporates the information in the data $\mathbf{X}$. So instead of finding a good prior on $\mathbf{w}$ directly, we first introduce a latent vector to $\mathbf{w}$, for which it is relatively easy to assign a prior that contain the data information. Specifically, we introduce a latent vector $\boldsymbol{y} = [y_1,\ldots,y_{n+m}]^{\mathrm{T}}$:

$$y_i = \mathbf{w}^T \phi(\mathbf{x}_i)$$

where y_i can be viewed as a soft label for the data point $\mathbf{x}_i$ and can be converted into the hard label t_i through the link function $\Theta(\cdot)$. Setting $\mathbf{H} = [\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_{n+m})]$ yields

$$\mathbf{y} = \mathbf{H}^T \mathbf{w} \tag{2}$$

If we give a prior on the label $\mathbf{y}$ conditional on the data $\mathbf{X}$, we can then transform the prior $p(\mathbf{y} \mid \mathbf{X})$ to the prior $p(\mathbf{w} \mid \mathbf{X})$ on the classifier $\mathbf{w}$.

Intuitively, we want the prior $p(\boldsymbol{y} \mid \mathbf{X})$ to impose a smoothness constraint on the soft labels and to encourage similar labels between similar data points. Inspired by graph regularization (Zhou *et al*., 2004) we use similarity graphs and their transformed Laplacian to induce priors on the soft labels $\mathbf{y}$.

To construct the prior $p(\mathbf{y} \mid \mathbf{X})$, we first form an undirected similarity graph over the data points. The data points are the nodes of the graph and the edge-weights between the nodes are based on similarity. This similarity is usually captured using a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. Examples of kernels include Gaussian and polynomials kernels. For Gaussian kernels, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$ where the kernel width σ controls the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Given the dataset $\mathbf{X}$ and a kernel, we can construct an $(n+m)\times(n+m)$ kernel matrix $\mathbf{K}$, where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ for all $i, j \in \{1, \ldots, n+m\}$. Note that the kernel matrix for semi-supervised learning involves both labeled and unlabeled data points. This is different from SVM kernels, which are defined by labeled data points only.

Given the similarity graph, we transform the kernel matrix $\mathbf{K}$ associated with the graph into the combinatorial Laplacian or the normalized Laplacian. Let us construct a matrix $\tilde{\mathbf{K}}$ the same as the matrix $\mathbf{K}$, except that the diagonal elements of $\tilde{\mathbf{K}}$ are set to zero, and define a diagonal matrix $\mathbf{G}$ where $\mathbf{G}_{ii} = \sum_j \tilde{\mathbf{K}}_{ij}$. The combinatorial Laplacian Δ and the normalized Laplacian $\tilde{\Delta}$ of the graph are defined as

$$\Delta = \mathbf{G} - \tilde{\mathbf{K}} \tag{3}$$

$$\tilde{\Delta} = \mathbf{I} - \mathbf{G}^{-\frac{1}{2}}\tilde{\mathbf{K}}\mathbf{G}^{-\frac{1}{2}} \tag{4}$$

where $\mathbf{I}$ is the identity matrix. Both the Laplacians are symmetric and positive semidefinite. For brevity, we slightly abuse the notation by using Δ for both the Laplacians. The construction of these Laplacian matrices are based on graph regularization theories. We impose a regularizer preferring soft labeling for which the norm $\mathbf{y}^{\mathrm{T}}\Delta\mathbf{y}$ is small. In a Bayesian framework, we assign a Gaussian prior distribution on y:

$$p(\mathbf{y} \mid \mathbf{X}) \propto e^{-\frac{1}{2}\mathbf{y}^{\mathrm{T}}\Delta_{\mathrm{y}}} \propto \mathcal{N}(\mathbf{y} \mid 0, \Delta^{-1}) \tag{5}$$

where $\mathcal{N}(\cdot \mid 0, \Delta^{-1})$ denotes a Gaussian probability function with mean 0 and variance Δ^{-1}. We can adjust the Laplacian matrices by changing their eigen-spectrum. Here, we use the normalized Laplacian matrices and add diagonal matrices with small values to them, avoiding the matrix inversion singularity.

Given the Gaussian prior on the labels $\mathbf{y}$, we construct the prior on the classifier w as follows:

$$\Sigma = (\mathbf{H}^{\mathbf{T}})^{-1}\Delta^{-1}(\mathbf{H}^{\mathbf{T}})^{-1} \tag{6}$$

$$p(\mathbf{w} \mid \mathbf{X}) = N(\mathbf{w} \mid 0, \Sigma) \tag{7}$$

where $(\mathbf{H}^{\mathbf{T}})^{-1}$ is the pseudo-inverse of $\mathbf{H}^{\mathbf{T}}$. This prior $p(\mathbf{w} \mid \mathbf{X})$ is consistent with the prior $p(\mathbf{y} \mid \mathbf{X})$ under the constraint between $\mathbf{y}$ and $\mathbf{w}$, i.e., $\mathbf{y} = \mathbf{H}^{\mathrm{T}}\mathbf{w}$. Again, we add some small positive values to the diagonal elements of Σ to enhance its stability.

2.2 Modeling probe confidence by likelihood

Assuming conditional independence of the observed labels, we have the factorized likelihood function $p(\mathbf{t}_L \mid \mathbf{y}) = \prod_{i=1}^{n} \Theta(t_i \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i))$. The likelihood function $\Theta(t_i\mathbf{x}_i^{\mathrm{T}}\mathbf{w})$ for each data point models the probabilistic relation between the observed label t_i and the input feature vector $\phi(\mathbf{x}_i)$. Gene expression datasets often contain noise, which may lead to labeling errors. Also, the qualities of different probes may vary. To model the probe confidence, we adopt the following flipping-error likelihood:

$$\begin{aligned}\Theta(t_i\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i)) &= \epsilon_i(1 - \mathrm{step}(t_i\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i))) \\ &\quad + (1 - \epsilon_i)\mathrm{step}(t_i\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i)) \\ &= \epsilon_i + (1 - 2\epsilon_i)\mathrm{step}(t_i\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i))\end{aligned} \tag{8}$$

where $\mathrm{step}(\cdot)$ is a step function such that $\mathrm{step}(t_i\mathrm{w}^{\mathrm{T}}\phi\mathbf{x}_i)) = 1$ if $t_i\mathrm{w}^{\mathrm{T}}\phi(\mathbf{x}_i) >= 0$ and $\mathrm{step}(t_i\mathrm{w}^{\mathrm{T}}\phi(\mathbf{x}_i)) = 0$ if $t_i\mathrm{w}^{\mathrm{T}}\phi(\mathbf{x}_i) < 0$, and ϵ_i models the uncertainty from the noise. This admits labeling errors with probability $\{\epsilon_i\}$. In our dataset, we have multiple probes that correspond to the same gene. The probe that is the closest to the most 3′ end of a gene more accurately measures the expression level of the given gene than the other probes, because the reverse transcription and amplification procedures introduce a bias against probes that are further away from the 3′ end. To model this effect, we set

$$\epsilon_{\mathrm{i}} = \begin{cases} e_l & \text{if probe i is most } 3' \\ e_h & \text{if probe i is not most } 3' \end{cases}$$

where $e_l > e_h$. By doing so, we give non-3′ probes a higher error rate than 3′ probes. Since this likelihood (8) explicitly models the labeling error rate, the model should be more robust to the presence of labeling noise in the data.

2.3 Computing the classifier posterior

Given the prior and the likelihood, the classifier posterior is

$$p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}_L) \propto p(\mathbf{w} \mid \mathbf{X}_L, \mathbf{X}_U) \prod_{i=1}^{n} \Theta(t_i\mathbf{x}_i^{\mathrm{T}}\mathbf{w}) \tag{9}$$

Because of the nonlinear likelihood terms, we can not compute the exact posterior in a closed form. Instead of using computationally expensive Monte Carlo methods, we apply an efficient deterministic Bayesian approximation technique, expectation propagation (EP) (Minka, 2001; Qi, 2004), to obtain a Gaussian approximation of the posterior $p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}_L)$. By exploiting the multiplication form (9) of the posterior, we iteratively refine the approximation of each likelihood term, eventually achieving an accurate approximate posterior. The algorithmic details for EP approximation of Gaussian classifiers can be found in Minka (2001).

2.4 Computing and combining probe predictions

As mentioned before, multiple probes are used to measure the expression levels of the same gene in the dataset we analyze. BGEN can classify each probe based on the classifier posterior $p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}_L)$. To combine multiple probe predictions, we use a soft decision procedure. Instead of simply averaging the binary probe classification results, we compute the predictive posterior probability for each probe and average these predictive posteriors for all corresponding probes to obtain the prediction for each gene. Specifically, given the approximate classifier posterior $p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}_L) \equiv \mathcal{N}(\mathbf{w} \mid \mathbf{m}_w, \mathbf{V}_w)$, where $\mathbf{m}_w$ and $\mathbf{V}_w$ are obtained from the EP approximation, we compute the predictive posterior for a probe as follows:

$$p(t_i \mid \mathbf{X}, \mathbf{t}_L) = \int p(t_i \mid \mathbf{w})p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}_L)\mathrm{d}\mathbf{w} \tag{10}$$

$$= \epsilon_i + (1 - 2\epsilon_i)\varphi(z) \tag{11}$$

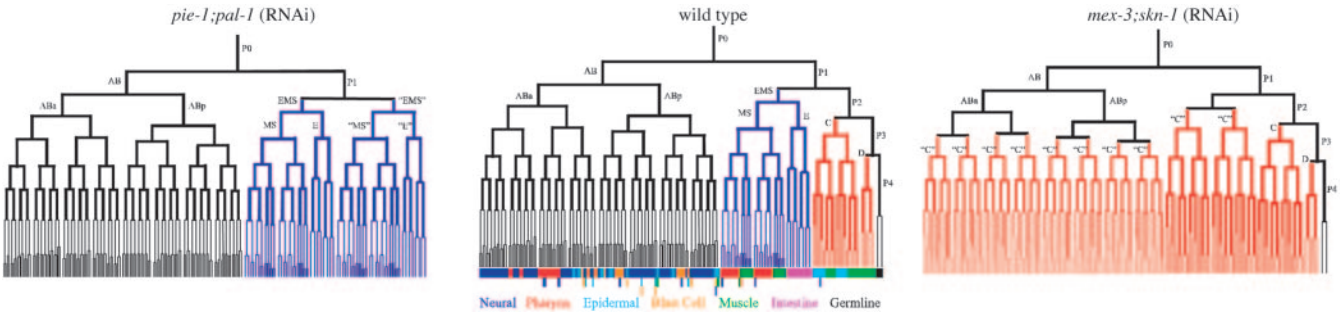


Fig. 1. Use of wild-type and mutant embryos to identify genes enriched in C lineage (adapted from Baugh *et al*. (2005)). Cell lineages are illustrated for wild-type embryos (middle), embryos of *pie-1*;*pal-1* (RNAi) genotype (left), and embryos of *mex-3*;*skn-1* (RNAi) genotype (right). C and EMS lineages are shown in red and purple, respectively.

where

$$z = \frac{t_i \phi(\mathbf{x}_i)^{\mathrm{T}} \mathbf{m}_w}{\sqrt{\phi(\mathbf{x}_i)\mathbf{V}_w \phi(\mathbf{x}_i)}} \tag{12}$$

and $\varphi(\cdot)$ is the cumulative distribution function of a Gaussian with mean 0 and variance 1. Equation (12) shows that the predictive posterior is controlled not only by the posterior mean $\mathbf{m}_w$ of the classifier, but also by the uncertainty, the variance $\mathbf{V}_w$ for the trained classifier. We average the predictive posteriors of the probes corresponding to the same gene k to obtain a gene predictive probability $p(\text{gene}_k \mid \mathbf{X}, \text{t}_L)$. Note that non-3′ probes contribute less to the gene prediction, since with a larger ϵ_i their predictive posteriors are less informative than the predictive posteriors of 3′ probes.

2.5 Automatic hyperparameter tuning

BGEN has a few hyperparameters, including kernel width σ and probe confidence levels e_l and e_h. To achieve a good test performance, we need to tune these hyperparameters. Here we adopt an automatic procedure to estimate them in a principled way. As a side-product of EP for our Bayesian learning, we estimate the approximate-leave-one-out error count or probability without carrying out leave-one-out cross-validation. The details can be found in Qi *et al*. (2004). We use the approximate leave-one-out error probability to estimate these hyperparameters.

2.6 Experimental validation of gene expression patterns

We examine gene expression patterns by using a reporter assay. We fuse selected gene promoters to yellow fluorescence protein (YFP) and a dominant *rol-6* gene by PCR (Hobert, 2002). 5′ genomic sequences up to the next upstream gene are used as promoters. YFP is amplified from pPD132.112 (Fire *et al*., 1990). The *rol-6* gene, a co-transformation marker, is amplified from pRF4 (Mello *et al*., 1991). Transgenic lines are obtained by injection of the reporter constructs. Chromosomal integration is performed by gamma irradiation. Using fluorescence microscopes we observe expression patterns of reporter genes in embryos from integrated transgenic lines.

3 RESULTS

This section describes the expression profile dataset used for our task, presents our prediction results for genes enriched in the C lineage, and compare the prediction accuracy of BGENs with those of K-means and SVMs. Finally, we confirm some predictions with biological experiments.

3.1 Summary of expression dataset

Baugh *et al*. (2005) profiled global gene expression for wild-type *C. elegans* embryos and two types of mutant embryos at 0, 23, 41, 53, 66, 83, 101, 122, 143, and 186 minutes after 4-cell stage. Embryos of the *pie-1*;*pal-1* (RNAi) genotype lack C-lineage cells, while embryos of the *mex-3*;*skn-1* (RNAi) genotype bear excess C-lineage cells (Figure 1).

Expression patterns of selected reporter genes in *C. elegans* embryos reflected whether these candidates were specific to the C lineage, and the confirmed candidates could be further classified as epidermis or muscle enriched (Baugh *et al*., 2005). Among the 40 candidates tested, 25 were confirmed to be C-lineage enriched. A non-specific gene list comes from an RNAi screen that identified 661 genes required for the first two cell divisions of the *C. elegans* embryo (Sonnichsen *et al*., 2005). The first two cell divisions occur well before the development of C lineage and these genes are believed to encode proteins for the basic mitotic machinery. Therefore, these genes are likely not to be specific to any lineage development.

3.2 Semi-supervised learning and comparison with K-means clustering and SVM classification

We use experimentally confirmed C-lineage genes reported by Baugh *et al*. (2005) as labeled positive examples, and use the non-specific genes required for early cell divisions as labeled negative examples.

For each gene, we calculate the difference of its expression levels in *mex-3*;*skn-1* (RNAi) embryos and *pie-1*;*pal-1* (RNAi) embryos at each time point, and use the ratios of this difference over the expression level in wild-type embryos as extracted features for clustering and classification. The maximum value of the ratios during development is also used as an extracted feature.

We compare BGEN with K-means clustering and SVM classification. First, we perform K-means clustering, which does not use the labeled information at all. The performance of K-means depends on the number of clusters which is unknown a priori. We use Silhouette scores to determine the optimal number of clusters (Kaufman & Rousseeuw, 1990). The Silhouette scores measure the tightness of a cluster and the separation of the given cluster from other clusters. More specifically, the Silhouette scores show how close a data point in one cluster is to data points in the neighboring clusters. The score ranges from +1, indicating that data points in one cluster are close to one another and are distant from data points in neighboring clusters, to −1, indicating the opposite. We compute the average Silhouette scores for all genes in the dataset. K-means with 2 clusters has the highest average score 0.8481. This score suggests that the two clusters obtained by K-means are coherent among themselves and well-separated from each other. To evaluate the capability of K-means to detect C-lineage genes, we designate a cluster to be C-lineage cluster if the ratio of labeled C-lineage genes to all genes in that cluster exceeds a specified threshold between 0 and 1; otherwise we designate it as a non C-lineage cluster. Genes in a C-lineage cluster are predicted to be C-lineage genes, and vice versa. We vary the threshold value and average the detection results over 200 runs with random initializations. The Receiver Operating Characteristic (ROC) curve from the averaged detection results is shown in Figure 2. K-means clustering performs poorly in terms of detecting C-lineage genes, though the clustering achieves a high average Silhouette score. The underlying reason may come from the fact that K-means clustering ignores any prior biological knowledge and purely depends on the expression dataset, and that C-lineage expression profiles are diverse.

For BGEN and SVM, we use experimentally confirmed C-lineage genes reported by Baugh *et al*. (2005), excluding genes used as positive training data, to evaluate the sensitivity. We use the non-specific genes required for early cell divisions, excluding genes used as negative training data, to assess the specificity.

For SVM training, we construct a pool of representative positive labels: *pal-1*, *vab-7*, *cwn-1*, *elt-1*, *elt-3*, *mab-21*, *hnd-1* and *hlh-1*. Each time 4 genes are randomly selected from this pool and serve as positive training examples. We randomly select 20 genes as negative training examples from the non-specific genes. We test the SVM prediction performance on the rest of the labeled data points. For BGEN, we use the same labeled examples, as well as about 900 unlabeled examples for training. We repeat this training and prediction procedure 10 times. We use Gaussian kernels for both SVM and BGEN. The regularization and kernel widths of SVM are tuned by leave-one-out cross-validations. For BGEN, both the kernel width and probe confidence levels are tuned based on the approximate leave -one-out error probability without actually carrying out leave-one-out cross -validations, as described in section 2.5. Based on the averaged prediction results, we plot ROC curves for BGEN and SVM (Figure 2). Overall BGEN performs significantly better than SVM. For example, with the same 80% specificity (i.e.,20% false positive rate), BGEN achieves 99% sensitivity (i.e., true positive rate), while SVM achieves only 82% sensitivity. Moreover, BGEN clearly outperforms K-means clustering in terms of detecting C-lineage genes as shown in Figure 2.

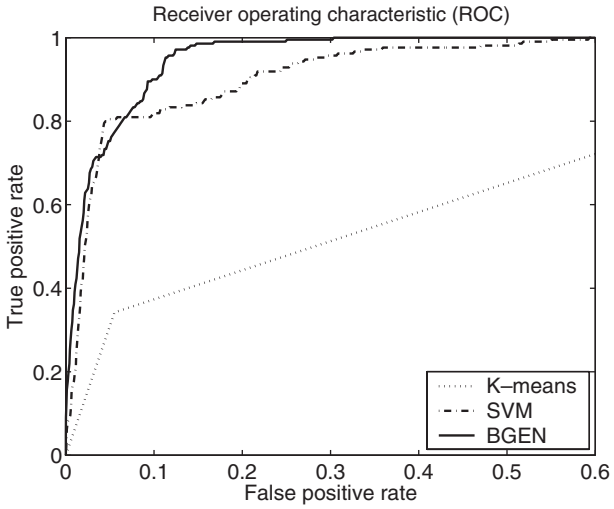


Fig. 2. Receiver Operating Characteristic (ROC) curves of BGEN, SVM, and K-means. Our semi-supervised learning method BGEN outperforms both SVM and K-means.

3.3 Whole genome prediction of C-lineage genes

Having tested the efficacy of BGEN, we predict C-lineage genes in the whole genome. We use 20 negative examples and all positive examples except for *pal-1*, because *pa1-1* is a maternally-supplied regulator while we are interested in identifying genes which are active in zygotic transcription during development. With 97% specificity evaluted by the non-specific gene set, we predicted 317 genes as enriched in C lineage, in addition to the previously confirmed C-lineage genes.

Our whole genome prediction is highly efficient in the sense that we use a kernel classifier pre-trained in a semi-supervised fashion to classify whole genome. This is different from many previous semi-supervised learning methods (Joachims, 2003; Zhou *et al*., 2004; Zhu *et al*., 2003), where either a re-training or a simple nearest-neighbor classifier is needed to classify new data points in addition to the training set.

BGEN may reduce potential false-positives from the original analysis. For example, F45E4.9(*hmg-5*), a HMG-box containing protein, which was previously predicted to be enriched in C lineage while our method classifies it as a non-C- lineage gene with a probabilistic confidence of 0.10. The experimental result showed that the expression pattern of F45E4.9 is not specific to the C lineage. This is also consistent with other reports in the literature that F45E4.9 is ubiquitously expressed in *C. elegans* embryos (Im & Lee, 2003). Another example is Y71F9AL.17, an uncharacterized gene that may be involved in intracellular trafficking and vesicular transport. Y71F9AL.17 was previously identified as a C-lineage candidate gene. In our analysis this gene receives a probabilistic confidence of 0.46 and is classified as non-C-lineage (Figure 3). The result of biological experiment was consistent with our prediction.

To visualize our predictions, we plot representative expression profiles for C-lineage genes and non-C-ineage genes with high prediction confidence (Figure 3). D1005.2 and F54D7.4 (the first column), two high-confidence C-lineage genes, are up- regulated

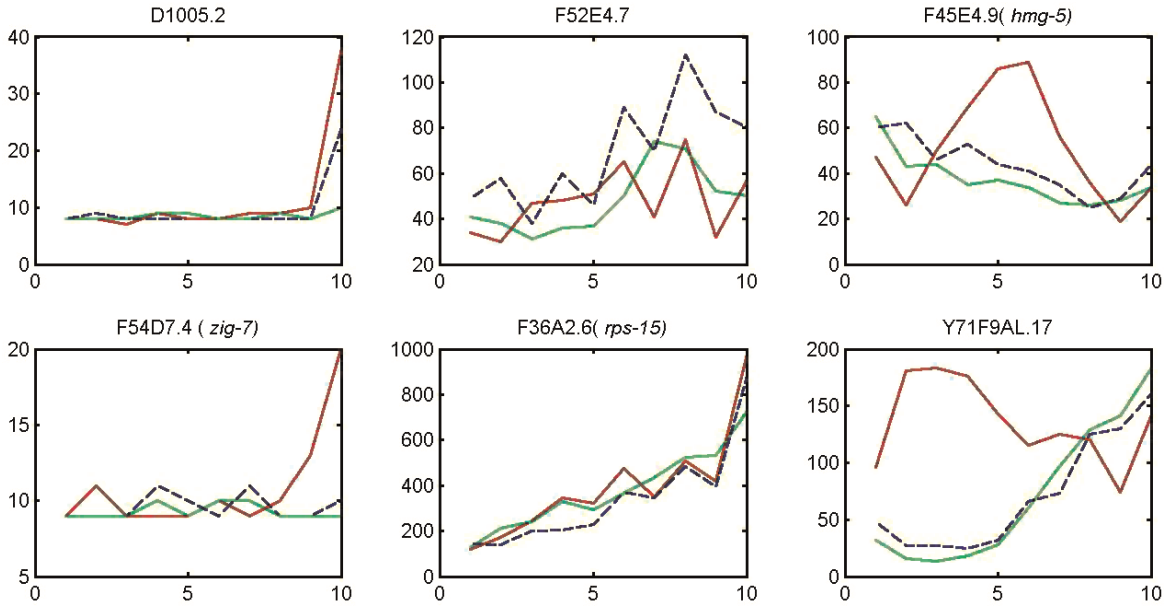


Fig. 3. Expression profiles of prediction examples. Red lines represent expression profiles in *mex-3*;*skn-1* (RNAi) embryos. Green lines represent expression profiles in *pie-1*;*pal-1* (RNAi)embryos. Blue dotted lines represent expression profiles in wild-type embryos. D1005.2 and F54D7.4 are high-confidence predictions of C-lineage genes. They receive confidence scores of 0.99 and 0.98, respectively. F52E4.7 and F36A2.6 are high-confidence predictions of non-C-lineage genes. They both receive confidence scores of 0.01. F45E4.9 and Y71F9AL.17 are less obvious examples. They receive confidence scores of 0.10 and 0.46, respectively, and are classified as non-C-lineage specific genes. Baugh *et al*. (2005) identified F45E4.9 and Y71F9AL.17 as C -lineage genes in their data analysis, but subsequent experimental results showed that these two genes were not specific to the C lineage.

in *mex-3*; *skn-1* (RNAi) embryos during development. F52E4.7 and F36A2.6 (the second column), two high-confidence non-C-lineage genes, do not exhibit such up-regulation of expression. The two examples of false-positives (F45E4.9 and Y71F9AL.17) by the previous analysis are also plotted. These two genes are prone to misprediction since they are up- regulated in *mex -3*; *skn-1* (RNAi) embryos. These examples illustrate the capability of BGEN to distinguish C-lineage genes from non-C-lineage genes even in some subtle cases.

3.4 Predictions of C epidermis and C muscle genes

During embryonic development, C-lineage cells differentiate into epidermis and muscle cells. Epidermis and muscle enriched genes are likely to exhibit slightly different expression profiles in wild-type and mutant embryos. Given our whole genome predictions of C-lineage genes, we apply BGEN to further distinguish the C- lineage genes as epidermis or muscle enriched. Baugh *et al*. (2005) showed by reporter assay that among the confirmed C-lineage genes, 15 were specifically expressed in epidermis cells and 4 were specifically expressed in muscle cells. We use this information to train and evaluate K-means, SVM, and BGEN. In addition to the normalized features used in 3.3, 2-level Daubechies wavelet decomposition of the difference features that explicitly represents the temporal and frequency information in the data is also computed as features.

Similar to what we have done before, we use the Silhouette scores to determine the number of clusters for K-means. For SVM and BGEN, we randomly select 6 epidermis and 2 muscle-genes and use them as training data. We use the rest of experimentally confirmed genes as the test set, which includes 9 epidermis genes and 2 muscle genes for each run. We repeat this procedure 5 times.

We evaluate the average area under the ROC curves for these three methods. For K-means, we compute the ROC curve using the same method as in the previous section. The average area under the ROC curve of BGEN is 0.80, indicating its prediction potential. The average areas achieved by K-means and SVM are only 0.56 and 0.50 respectively, indicating the failure of the K-means and SVM predictions. This further demonstrates the advantage of our semi- supervised learning method. For the run in which BGEN achieves the largest area under the ROC curve, we correctly predict all 9 epidermis genes and 2 muscle genes in the test set. The prediction accuracy achieved by BGEN suggests the epidermis genes and muscle genes may be separable from each other in terms of expression profiles. However, this prediction accuracy should not be over- interpreted, because both the training and testing datasets are small. In the future, more labeled data and additional microarray datasets may be integrated to improve the predictions. The lists of predicted C epidermis and C muscle genes can be downloaded at http://www.csail.mit.edu/~alanqi/~projects/BGEN.html.

3.5 Experimental verification of predictions

We predict K01A2.5 and R11A5.4, two uncharacterized genes, as enriched in C lineage. These two genes were also identified in previous analysis as C-lineage candidates but were not tested (Baugh *et al*., 2005). We further identify K01A2.5 as epidermis enriched and R11A5.4 as muscle enriched. We examine their expression patterns by reporter assay. The expression patterns

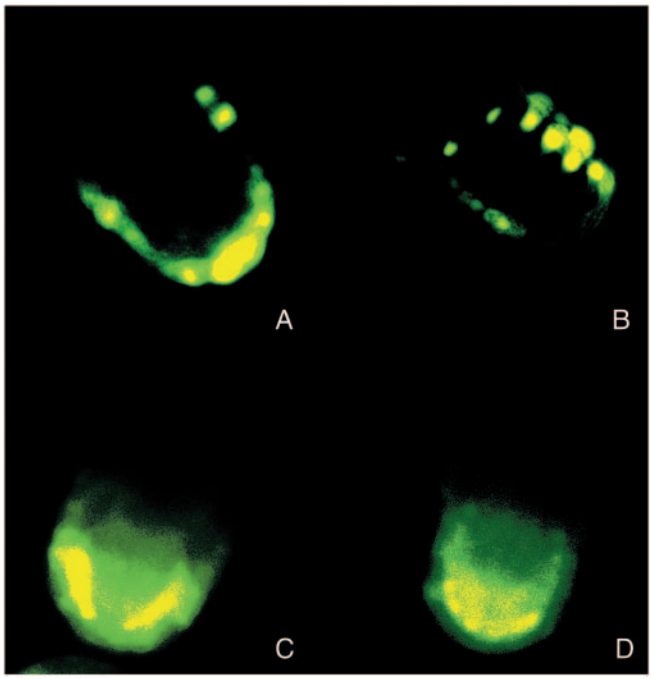


Fig. 4. Experimental validation of redictions. We predict K01A2.5 and R11A5.4, as enriched in C epidermis cells and enriched in C muscle cells, respectively. We examine expression patterns of K01A2.5 (A, B) and R11A5.4 (C, D) in developing *C. elegans* embryos. The experimental results are consistent with our predictions for both genes.

of reporter genes in *C. elegans* embryos are consistent with our predictions (Figure 4). The reporter gene that contains K01A2.5 promoter is expressed in C epidermis cells, and the reporter gene that contains R11A5.4 promoter is expressed in C muscle cells. The experimental results support that our methodology yields relevant biological insights to elucidate developmental processes.

4 CONCLUSIONS

We have developed BGEN, a novel semi-supervised learning method, which utilizes both large-scale expression datasets and prior biological knowledge to identify target genes. Using BGEN, we have predicted genes enriched in C lineage during *C. elegans* embryonic development, and have further classified C-lineage candidate genes according to tissues where they are enriched. In comparison with unsupervised K-means clustering and supervised SVM classification, our semi-supervised learning method achieves higher sensitivity and specificity. We experimentally confirm two examples from our predictions, which further supports our methodology. As a powerful computational tool, BGEN can be used to refine target selection from large-scale expression datasets for many other biological problems in the future.

ACKNOWLEDGEMENTS

We thank R. Dowell for critical reading of our manuscript. H.G. is supported by Whitehead Institute and supported in part by NIH GM 644429 to C.P.H.

REFERENCES

Baugh,L.R., Hill,A.A., Claggett,J.M., Hill-Harfe,K., Wen,J.C., Slonim,D.K., Brown,E.L. and Hunter,C.P. (2005) The homeodomain protein PAL-1 specifies a lineage-specific regulatory network in the *C. elegans* embryo. *Development*, **132** (8), 1843–1854.

Belkin,M. and Niyogi,P. (2004) Semi-supervised learning on Riemannian manifolds. *Machine Learning, Special Issue on Clustering*, **56**.

Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA*, **95** (25), 14863–8.

Fire,A., Harrison,S.W. and Dixon,D. (1990) A modular set of lacz fusion vectors for studying gene expression in *Caenorhabditis elegans*. *Gene*, **93** (2), 189–98.

Furlong,E.E.M., Andersen,E.C., Null,B., White,K.P. and Scott,M.P. (2001) Patterns of gene expression during *Drosophila* mesoderm development. *Science*, **293** (5535), 1629–1633.

Gaudet,J. and Mango,S.E. (2002) Regulation of organogenesis by the *Caenorhabditis elegans* FoxA protein PHA-4. *Science*, **295** (5556), 821–825.

Hobert,O. (2002) PCR fusion-based approach to create reporter gene constructs for expression analysis in transgenic *C. elegans*. *Biotechniques*, **32** (4), 728–30.

Im,S.H. and Lee,J. (2003) Identification of HMG-5 as a double-stranded telomeric DNA-binding protein in the nematode *Caenorhabditis elegans*. *FEBS Lett*, **554** (3), 455–61.

Joachims,T. (2003) Transductive learning via spectral graph partitioning. *ICML*.

Kaufman,L. and Rousseeuw,P.J. (1990) *Finding Groups in Data: an Introduction to Cluster Analysis*. Wiley.

Mello,C.C., Kramer,J.M., Stinchcomb,D. and Ambros,V. (1991) Efficient gene transfer in *C. elegans*: extrachromosomal maintenance and integration of transforming sequences. *Embo J*, **10** (12), 3959–70.

Minka,T.P. (2001) Expectation propagation for approximate Bayesian inference. In *Uncertainty in Artificial Intelligence*.

Qi,Y. (2004) *Extending expectation propagation for graphical models*. Ph.D. thesis, MIT.

Qi,Y., Minka,T.P., Picard,R.W. and Ghahramani,Z. (2004) Predictive automatic relevance determination by expectation propagation. In *Proceedings of Twenty-first International Conference on Machine Learning*.

Reinke,V., Smith,H.E., Nance,J., Wang,J., Van Doren,C., Begley,R., Jones,S.J.M., Davis,E.B., Scherer,S., Ward,S. and Kim,S.K. (2000) A global profile of germline gene expression in *C. elegans*. *Molecular Cell*, **6** (3), 605–616.

Robertson,S.M., Shetty,P. and Lin,R. (2004) Identification of lineage -specific zygotic transcripts in early *Caenorhabditis elegans* embryos. *Dev Biol*, **276** (2), 493–507.

Sonnichsen,B., Koski,L.B., Walsh,A., Marschall,P., Neumann,B., Brehm,M., Alleaume,A.M., Artelt,J., Bettencourt,P., Cassin,E., Hewitson,M., Holz,C., Khan,M., Lazik,S., Martin,C., Nitzsche,B., Ruer,M., Stamford,J., Winzi,M., Heinkel,R., Roder,M., Finell,J., Hantsch,H., Jones,S.J., Jones,M., Piano,F., Gunsalus,K.C., Oegema,K., Gonczy,P., Coulson,A., Hyman,A.A. and Echeverri,C.J. (2005) Full-genome RNAi profiling of early embryogenesis in *Caenorhabditis elegans*. *Nature*, **434** (7032), 462–9.

Sulston,J.E., Schierenberg,E., White,J.G. and Thomson,J.N. (1983) The embryonic cell lineage of the nematode *Caenorhabditis elegans*. *Dev Biol*, **100** (1), 64–119.

Szummer,M. and Jaakkola,T. (2003) Information regularization with partially labeled data. *NIPS*.

Zhou,D., Bousquet,O., Lal,T.N., Weston,J. and Scholkopf,B. (2004) Learning with local and global consistency. *NIPS*, **16**.

Zhu,X., Ghahramani,Z. and Lafferty,J. (2003) Semi-supervised learning using Gaussian fields and harmonic functions. *ICML*.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e424–e430
doi:10.1093/bioinformatics/btl254

Decoding non-unique oligonucleotide hybridization experiments of targets related by a phylogenetic tree

Alexander Schliep[1,*] and Sven Rahmann[2,3,*]

[1]Dept. Computational Molecular Biology, Max Planck Institute for Molecular Genetics, Ihnestraße 63–73, D-14195 Berlin, Germany, [2]Algorithms and Statistics for Systems Biology group, Genome Informatics, Technische Fakultät, Universität Bielefeld, D-33594 Bielefeld, Germany and [3]International NRW Graduate School of Bioinformatics and Genome Research, Universität Bielefeld

ABSTRACT

Motivation: The reliable identification of presence or absence of biological agents ("targets"), such as viruses or bacteria, is crucial for many applications from health care to biodiversity. If genomic sequences of targets are known, hybridization reactions between oligonucleotide probes and targets performed on suitable DNA microarrays will allow to infer presence or absence from the observed pattern of hybridization. Targets, for example all known strains of HIV, are often closely related and finding *unique* probes becomes impossible. The use of *non-unique* oligonucleotides with more advanced decoding techniques from statistical group testing allows to detect *known* targets with great success. Of great relevance, however, is the problem of identifying the presence of previously *unknown* targets or of targets that evolve rapidly.

Results: We present the first approach to decode hybridization experiments using non-unique probes when targets are related by a phylogenetic tree. Using a Bayesian framework and a Markov chain Monte Carlo approach we are able to identify over 94% of known targets and assign up to 70% of unknown targets to their correct clade in hybridization simulations on biological and simulated data.

Availability: Software implementing the method described in this paper and datasets are available from http://algorithmics.molgen.mpg.de/probetrees.

Contact: alexander.schliep@molgen.mpg.de, Sven.Rahmann@cebitec.uni-bielefeld.de

1 INTRODUCTION

Identifying biological targets. Identifying viruses infecting a patient, detecting bacteria spoiling food, or deciding whether a water sample is safe for humans to drink are tasks which share the same underlying problem: to identify certain *targets* in biological (DNA) samples. Targets refer to the biological agents, the viruses, bacteria or other organisms that we want to detect. Recent developments in the Avian influenza pandemic brought virus identification into the front-news spotlight. In addition to accurately determining the lethal virus strain [Putonti *et al*., 2006], it is crucial to screen humans and animals, which might host several viruses and thus allow cross-species recombination. More optimistic applications of target detection are the study of biodiversity, say on the microbial level, and environmental microbiology. The target identification problem is also central in the area of biothreat reduction.

In clinical applications, target identification has classically been achieved for individual targets with unique markers such as staining techniques for specific antibodies. While one test per potential target is acceptable for many medical applications, it is not a cost-effective strategy if the number of potential targets is large, if several targets might be present simultaneously, or if many samples must be investigated. In South Africa for example, HIV super-infections, i.e., simultaneous infections with multiple HIV strains, are much more prevalent than in the Western world. In these cases, clinical marker kits for strain identification are more prone to failure.

Approaches based on unique probes. One experimental assay widely used in molecular biology is the hybridization reaction of fluorescently labeled DNA or RNA molecules to complementary DNA or RNA. Such hybridization reactions can be used for target detection if (partial) genomic sequences of targets are available. Often, short oligonucleotide DNA microarrays are used as technology platform (the approach in principle generalizes to other hybridization-based technologies). Assuming ideal conditions, we would select one specific oligonucleotide probe that hybridizes to its intended target only and does not cross-hybridize to any other target. Subsequently, we detect presence and absence of targets in a sample from the observed hybridization pattern. This *unique probe* approach has been originally developed for the design of gene expression DNA microarrays using oligonucleotide probes (e.g., [Kaderali and Schliep, 2002; Rahman, 2003a]). However, in the applications described above, targets are often closely related and thus unique probes cannot be found.

Non-unique probes. The use of *non-unique probes*, hybridizing to several targets simultaneously, poses problems in the analysis of experiments. If one assumes that at most one target can be present simultaneously, the problem can be handled effectively [Wang *et al*., 2003, Rash and Gusfield, 2002]. This assumption is unrealistic, however, and [Schliep *et al*., 2003] introduced a *statistical group-testing* approach to address the case when multiple targets are present simultaneously. Subsequent work [Klau *et al*., 2004] has attempted to minimize the number of probes required to reliably identify small-cardinality target sets by an integer linear programming approach. In all of the above work, only the ability to detect *known* targets has been evaluated.

*To whom correspondence should be addressed.
Both authors contributed equally.

Novel contributions. We extend the group-testing approach using non-unique probes to targets related by a phylogenetic tree. This allows us to consider an intriguing and highly relevant question: Can we even detect the presence of yet *unknown* targets, e.g., can we detect the presence of a new strain, or can we detect the presence of a known target if it (and its hybridization pattern) has changed because of fast evolution? Even if we restrict ourselves to a specific virus, the targets used as input will only represent a sample of all existing strains and new strains are likely to arise between the time of microarray design and its large-scale use. To our knowledge, this article is the first work to address these issues.

Outline. We describe the probe selection strategy and group testing methods in Section 2, particularly focusing on the novel aspect how they can be integrated with phylogenetic tree information. Section 3 presents artificial and real datasets for evaluating these methods, describes our evaluation criteria, and shows the evaluation results. A concluding discussion is given in Section 4.

2 METHODS AND MODELS

Notational Remarks. If $\mathcal{S}$ is a finite set, $|\mathcal{S}|$ is its cardinality. We identify binary vectors $T \in \{0,1\}^m$ with the index set $\{i : T_i = 1\} \subset \{1,\ldots,m\}$, for which we also write T, so $|T| = \sum_{i=1}^{m} T_i$.

2.1 Overview: Problem setting

Initially, we are given a set $\mathcal{T}$ of DNA target sequences (the *known targets*) and a phylogenetic tree $\mathcal{B}$ relating them. Depending on the application, the targets might be whole genomes (e.g., all known HIV strain genomes), or single gene sequences (e.g., the cytochrome C sequences of several related species). We assume that the target set contains many closely related and hence similar sequences.

Our objective is to be able to decide which of these targets are present and which ones are absent in unclassified DNA samples when we observe an oligonucleotide probe hybridization fingerprint for the sample. To be more precise, we assume that we observe which probes react to some target(s) in the sample, but that this observation is *noisy*. In most applications, we may assume that the target set contained in the sample is small compared to the whole set $\mathcal{T}$ (e.g., the set of HIV strains infecting a single patient).

Additionally, we expect that the sample may contain *unknown targets*, that is, sequences similar to those in $\mathcal{T}$ that were not available when $\mathcal{T}$ was prepared. This would be the case for new virus strains or fast evolving genomes, for example. Although we cannot expect to perfectly classify these unknown targets, we would at least like to place them at the correct location in the tree $\mathcal{B}$.

Our first tasks are thus

(1) to select suitable *probe candidates* for the given target set $\mathcal{T}$. Note that the usual probe design methodologies that look for target-specific probes do not have a good chance of success on the typical datasets we consider: Because of the high sequence similarity between targets, only very few specific probes will be found. Our proposed solution is to use a *group testing* approach that allows *non-unique* probes. We deal with the ensuing complications in a subsequent *decoding* step. The candidate selection step also ensures that no probes are selected that could hybridize to genomes of contaminating organisms or host organisms (e.g., the human genome for HIV viruses);

(2) to reduce the candidate set to a final *probe set* $\mathcal{P}$;

(3) to compute the $|\mathcal{T}| \times |\mathcal{P}|$ *basic hybridization matrix* H^{basic}, a binary matrix defined by $H^{\text{basic}}_{ij} = 1$ if target i hybridizes to probe j, and $H^{\text{basic}}_{ij} = 0$ otherwise;

(4) to extend the hybridization patterns (rows) of H^{basic} from targets to whole subtrees (monophyletic groups) of $\mathcal{B}$ by deciding which hybridization pattern would be ''typical'' for unknown targets in a monophyletic group. We obtain an *(extended) hybridization matrix* H of size $(|\mathcal{T}| + |\mathcal{I}|) \times |\mathcal{P}|$, where $\mathcal{I}$ denotes the set of internal (non-leaf) nodes of $\mathcal{B}$.

The above steps are described formally in Sections 2.2 (probe selection) and 2.3 (computing H), followed by a small example.

Given H and a target set $T \subset \{1,\ldots,|\mathcal{T}| + |\mathcal{I}|\}$, it is straightforward to compute the theoretical (i.e., error-free) hybridization result $r = r(T) \in \{0,1\}^{|\mathcal{P}|}$: We will observe $r_j = 1$ if there exists a target $i \in T$ to which probe j hybridizes ($H_{ij} = 1$). In other words, $r_j = \vee_{i \in T} H_{ij}$, so r is the logical or of the rows indicated by T. In reality, however, we need to take noisy results into account: Probes not showing a hybridization signal although they should are called *false negatives*, and probes showing a signal although they should not are called *false positives*. The error model is described in Section 2.4.

For an unidentified DNA sample, we need to solve the inverse problem of the above one: We observe a certain result r, and our task is to find T, which may consist of both known targets $t \in \mathcal{T}$ and unknown targets $t \in \mathcal{I}$ modeled by internal nodes of $\mathcal{B}$, such that T best explains r. We adopt a Bayesian framework and introduce a target set prior in Section 2.5. Then our goal becomes to find the target set that maximizes the posterior probability given r, which turns out to be a difficult problem to solve exactly. We thus switch to a Gibbs sampling strategy, which we describe in Section 2.6.

2.2 Probe selection

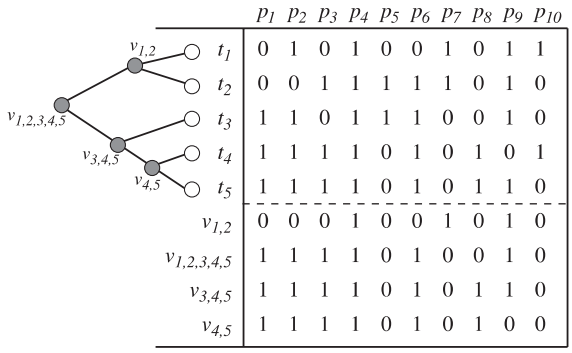
We start with a set $\mathcal{T} = \{t_1,\ldots,t_m\}$ of m distinct but similar DNA sequences, the *targets*. The first step is to find characteristic substrings (the *probes*) either for single targets or for whole target sets $T \subset \mathcal{T}$. The idea is that an unidentified DNA sample can be tested quickly and (relatively) cheaply for the occurrence of all probe sequences, e.g., by a microarray hybridization experiment, whereas determining the precise sequences of all sample members would be a more complicated procedure.

A good (specific or unique) probe p is characterized by the fact that it hybridizes well to a single target and not at all to the remaining targets. Because of the high sequence similarity in $\mathcal{T}$, however, unique probes will be difficult to find in sufficient number. Instead of attempting a bad compromise, we turn this problem into a feature and allow that p hybridizes to a small group $\mathcal{T}_p$ of targets; this need not be a monophyletic group in $\mathcal{B}$. We require, however, that the probe makes a clear distinction between $\mathcal{T}_p$ and $\mathcal{T} \setminus \mathcal{T}_p$ in the sense that there is a strong observable signal for all $t \in \mathcal{T}_p$ and no signal for all $t \in \mathcal{T} \setminus \mathcal{T}_p$.

The dynamics of DNA-DNA hybridization are quite complicated and not fully understood. However, it is reasonable to assume that a probe will give a clear positive signal if it is an exact substring of the target, and that no signal will be observed if the longest common substring between probe and target is very short. This so-called *longest common factor approach* was first proposed in [Rahmann, 2003a, 2002] and provides a practical and efficient surrogate measure for the true probe-target affinity. What must be avoided are probes that have long but not full-length common substrings with some targets in $\mathcal{T}$.

We thus proceed as follows. Every substring p in a given length range (our method is mainly applicable to short oligonucleotides between 20 and 30 nt) of any target in $\mathcal{T}$ is tested against the other targets for long (but not full-length) common substrings and discarded as a probe candidate if any are found. For the remainder, the hybridization stability (Gibbs free energy) is estimated using the nearest-neighbor model described in [SanataLucia, 1998]. The probes are accepted only if their estimated Gibbs free energy falls into a small homogeneous range to ensure similar hybridization behavior. All of these steps are implemented in the existing PROMIDE software described in [Rahmann, 2003a]. The main reason to choose PROMIDE is that it is one of the few programs that allows non-uniqe probe selection.

The nature of the selection process allows to model hybridization as a yes/no event that can be described by a binary matrix H^{basic}: Consider the



	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
t_1	0	1	0	1	0	0	1	0	1	1
t_2	0	0	1	1	1	1	1	0	1	0
t_3	1	1	0	1	1	1	0	0	1	0
t_4	1	1	1	1	0	1	0	1	0	1
t_5	1	1	1	1	0	1	0	1	1	0
$v_{1,2}$	0	0	0	1	0	0	1	0	1	0
$v_{1,2,3,4,5}$	1	1	1	1	0	1	0	0	1	0
$v_{3,4,5}$	1	1	1	1	0	1	0	1	1	0
$v_{4,5}$	1	1	1	1	0	1	0	1	0	0

	T	Hybridization result r
a.	$\{1\}$	0 1 0 1 0 0 1 0 1 1
b.	$\{1,2\}$	0 1 1 1 1 1 1 0 1 1
c.	$\{v_{1,2}\}$	0 0 0 1 0 0 1 0 1 0
d.	$\{v_{1,2},3\}$	1 1 0 1 1 1 1 0 1 0
e.	$\{2,4\}$	1 1 1 1 1 1 1 1 1 1
f.	$\{1,2,5\}$	1 1 1 1 1 1 1 1 1 1
g.	???	1 0 1 0 1 0 1 1 0 1

Fig. 1. *Left*: A small hybridization matrix H. Rows 1–5 define a hypothetical *basic hybridization matrix* H^{basic}, as it would result from a probe selection process. Rows 6–9 are associated to the internal nodes of the phylogenetic tree $\mathcal{B}$ shown to the left of H^{basic}. They are computed as strict majority functions and represent any so far unknown target that could exist in the monophyletic group below the respective node. *Right*: Seven examples (a–g) of sets of known and unknown targets and their expected hybridization results (the or of the rows indicated by the target set); see Section 2.4 for details.

relation of target i and probe candidate j: Either the probe is a substring of the i (in which case we assume a stable hybridization and set $H^{\text{basic}}_{ij} := 1$), or they share only a short common substring (in which case stable hybridization does not occur and we set $H^{\text{basic}}_{ij} := 0$). The intermediate case where ''almost'' the whole probe occurs in some target is ruled out by the longest-common-factor-based selection process.

In the resulting set of probe candidates, maybe no probe identifies any target uniquely, but certain combinations of probes still identify certain combinations of targets (target sets). This places our approach in the field of *group testing*: Each probe tests whether *some* member of a certain target group is present, but cannot tell *which one*. The resulting decoding problem is described in Section 2.6.

Reducing the set of probe candidates. In previous work [Klau *et al*., 2004], we have shown that the resulting probe candidate set can often be reduced considerably (up to 50%) without sacrificing decoding resolution if the probes are picked carefully. However, in this study, we face a different task: One of our goals is to detect unknown or fast-evolving targets. Therefore, any optimization of the probe set, even if it does not adversely affect our ability to identify known targets, would certainly decrease our chances of identifying unknown targets. Therefore, we have not reduced the probe set.

2.3 Extending the hybridization matrix to monophyletic groups

After probe selection, we have n probes $\mathcal{P} = \{p_1, \ldots, p_n\}$ for the m targets $\mathcal{T} = \{t_1, \ldots, t_m\}$, we know the basic hybridization matrix H^{basic}_{ij}, as described above, and are given a phylogenetic tree $\mathcal{B}$ with the targets at the leaves and a set $\mathcal{I}$ of internal nodes defining monophyletic target groups.

Since we want to detect unknown targets $t \notin \mathcal{T}$ to a degree that we can place them at an approximately correct location in the phylogenetic tree $\mathcal{B}$, we need to model a ''typical'' hybridization pattern of an unknown target that belongs to each particular monophyletic group.

Let v denote an internal node in $\mathcal{B}$ and let $L(v) := \{i : t_i \text{ is a leaf below } v\}$ denote the set of target indices that form the monophyletic group below v. Our approach is to postulate that probe p_i is ''typical'' for v if it hybridizes to more than half of the targets in $L(v)$. We thus define the hybridization vector $h(v)$ by the strict majority function,

$$h(v) \equiv (h_{v,1}, \ldots, h_{v,n}) \in \{0,1\}^n \text{ with}$$
$$h_{v,j} := 1 \Longleftrightarrow \sum_{i \in L(v)} H^{\text{basic}}_{i,j} > |L(v)|/2.$$

One alternative would be to use the logical and function (i.e., set $h_{v,j} := \wedge_{i \in L(v)} H^{\text{basic}}_{ij}$), but intuitively this does not capture the ''typicality'' of probes as well as the majority function. Nevertheless, other alternatives are certainly possible; the aim being to guess as precisely as possible the hybridization behavior of unknown targets in a monophyletic group, which is *per se* an impossible task.

To build the extended hybridization matrix H of size $(m + |\mathcal{I}|) \times n$, we define the first m rows as those in H^{basic}. To define the remaining $|\mathcal{I}|$ rows, we assign numbers $i(v)$ ranging from $m+1$ to $m + |\mathcal{I}|$ bijectively to the internal nodes $v \in \mathcal{I}$ and define the $i(v)$-th row of H as the majority vector $h(v)$.

An example of an extended hybridization matrix H with 5 targets and 10 probes, along with the phylogenetic target tree $\mathcal{B}$ with 4 internal nodes, is shown in Figure 1 (left).

2.4 Probabilistic hybridization model

As stated in Section 2.1, the expected hybridization result $r = r(T)$ of a target set $T \subset \{1, \ldots, m + |\mathcal{I}|\}$ is obtained by computing the logical $\circ r$ of the indicated rows of the hybridization matrix H. It is understood that if I contains representations of unknown targets u (indices ranging from $m+1$ to $m + |\mathcal{I}|$), r is not the actual hybridization pattern of T, since the actual behavior of u is unknown and only hypothesized to look similar to the corresponding row in H.

As an example, consider Figure 1 (right). The expected result for singleton target sets can be read directly from H (examples a, c). If $|T| \geq 2$, the result is the logical $\circ r$ of the corresponding rows (examples b, d–f). The set $\{v_{1,2}\}$ represents a *single typical unknown* target somewhere below $v_{1,2}$ (and no further targets) and must be distinguished from $\{1,2\}$ that consists of *two particular known* targets (and no further targets). Target sets may mix known and unknown targets (example d). Sometimes, the same result may occur for several distinct target sets (examples e, f; there are many more target sets giving rise to this ''all ones'' result). Other results may not be explainable by any target set at all without allowing errors (example g).

In order to model false positive and false negative hybridizations, we switch to a probabilistic model, where r becomes a random vector whose distribution depends on T and the assumed error rates. We use a model with two error parameters: f_- denotes the (per probe and target) probability that a hybridization fails, and f_+ denotes the (per probe) probability that a probe shows a signal although no hybridization should take place. In practice, we must assume error rates of up to 0.1.

We define $\mathcal{P}_i := \{j \in \{1, \ldots, n\} : H_{ij} = 1\}$ as the set of probes hybridizing to target i, and $\mathcal{T}_j := \{i \in \{1, \ldots, m + |\mathcal{I}|\} : H_{ij} = 1\}$ as the set of targets hybridizing to probe j.

For given T, in order to observe no signal at probe j, *all* of the $|T \cap \mathcal{T}_j|$ expected hybridizations must fail. Assuming independence between these failures, this event occurs with probability $f_-^{|T \cap \mathcal{T}_j|}$. Additionally, the probe must not show a false positive reaction; this event has probability $1 - f_+$ and

is also assumed to be independent of potential failure events. It follows that

$$\eta_j(T) :\equiv \mathbb{P}(r_j = 0 \,|\, T) = f_-^{|T \cap \mathcal{T}_j|} \cdot (1 - f_+), \tag{1}$$

and that $\mathbb{P}(r_j = 1 \,|\, T) = 1 - \eta_j(T)$.

We further assume that all probes react independently, such that the joint probability that the observed result is a particular vector $r = (r_j)$ is given by the product

$$\mathbb{P}(r \,|\, T) = \prod_{j=1}^{n} (1-\eta_j(T))^{r_j} \cdot (\eta_j(T))^{1-r_j}. \tag{2}$$

For example, assuming $f_+ = f_- = 0.05$, the result $r = (1,0,1,0,1,0,1,1,0,1)$ in Figure 1 (Example g) has probability $2.1 \cdot 10^{-7}$ if $T = \{4\}$ and $1.3 \cdot 10^{-8}$ if $T = \{\}$.

As an example on a larger scale, consider error rates of 10% in an experiment with 1000 probes and a target set T with a single target covered by 10 probes. We expect one false negative, nine true positive and 100 false positive probes. Even though the number of false positives is much larger than the number of true positives, correct target identification will be possible in most cases because the false positive probes do not paint a consistent picture, while the true positive probes do.

2.5 Target set prior

To identify a DNA sample, we are given a realization of r and are asked for the target set T that best explains the observation. In principle, we could proceed by a maximum likelihood approach, i.e., attempt to find T^* that maximizes $\mathbb{P}(r \,|\, T)$ over all T . However, from the example in Figure 1, we see that this would cause problems for results such as $r = (1,1,\ldots,1)$ that have many good explanations. In accordance with our sparseness assumptions and Occam's razor, we prefer a parsimonious explanation (small $|T|$), but the likelihood model specified by Eqs. (1), (2) actually prefers larger target sets.

We thus move to a Bayesian framework and introduce a prior probability distribution on the potential target sets, defined by a ''prevalence'' vector $f = (f_1, \ldots, f_{m+|\mathcal{I}|}) \in [0, 1/2]^{m+|\mathcal{I}|}$, where f_i denotes the a-priori probability that target i is contained in T, and all target occurrences are assumed independent:

$$\mathbb{P}(T) = \prod_{i=1}^{m+|\mathcal{I}|} f_i^{T_i} \cdot (1 - f_i)^{1-T_i}. \tag{3}$$

The relative magnitude f_i/f_k determines how much more likely it is a-priori to see target i in an unclassified sample than target k. Such ratios are available for many applications, e.g., the relative prevalences of HIV subtypes in patients. If nothing is known, a flat prevalence prior may be used where all f_i are equal. The absolute magnitude $F = \sum_i f_i$ should be chosen such that $f_i \ll 1/2$ for all i, and depending on how many probes are available to decide reliably on inclusion or exclusion of target i. In practice, we recommend $f_i \approx 0.01$ to favor non-inclusion of each target 99-fold over its inclusion a-priori.

2.6 Decoding hybridization results

Maximum a-posteriori. By Bayes Theorem, the posterior probability of a target set T given a hybridization result r is

$$\mathbb{P}(T \,|\, r) = \frac{\mathbb{P}(T) \cdot \mathbb{P}(r \,|\, T)}{\mathbb{P}(r)} \propto \mathbb{P}(T) \cdot \mathbb{P}(r \,|\, T), \tag{4}$$

where $\mathbb{P}(r)$ is a constant. We are interested in finding sets $T \subset \{1, \ldots, m + |\mathcal{I}|\}$ that explain r well in the sense that $\mathbb{P}(T \,|\, r)$ is high. For very small examples, such as the one in Figure 1, we can compute the posterior for all T directly and find the maximizing set T^* by brute force. For example, assuming error rates $f_+ = f_- = 0.05$ and prior prevalences $f_i = 0.33$ for all i, the two best explanations for the observation r in Figure 1 (Example g) are $T_1 = \{4\}$ with $\mathbb{P}(T_1 \,|\, r) = 0.775$ and $T_2 = \{\}$ with $\mathbb{P}(T_2 \,|\, r) = 0.094$.

However, since $\mathbb{P}(T \,|\, r)$ is a complicated function of T, direct maximization seems out of reach for realistically large datasets. Additionally, there may be several good distinct solutions.

Posterior marginals. For the above reasons, instead of maximizing the posterior, we estimate the *posterior marginals* $\mu_i := \mathbb{P}(T_i = 1 \,|\, r)$ and the *posterior target set cardinality* $M := \mathbb{E}[|T| \,|\, r] = \sum_i \mu_i$ to decide how many and which targets are the best candidates for explaining r. In the toy example, we find that $\mu_4 = 0.81$ and $\mu_2 = 0.06$ are the highest posteriors and $M = 0.95$ indicates that we expect slightly less than one target to be present.

In larger problems, we estimate these quantities by Gibbs sampling from the posterior. The next paragraphs show that this can be done efficiently in our model.

Gibbs sampling. In our setting, Gibbs sampling consists of a pre-defined number of rounds, during each of which we update the target set T, which is initially random. Each round consists of $m + |\mathcal{I}|$ steps, and in step i of each round we decide whether target t_i should be included in or removed from T by considering the posterior ratio $\rho \equiv \rho_i(T)$ defined as follows: If $i \notin T$, let $T^+ := T \cup \{i\}$, otherwise, if $i \in T$, let $T^- := T \backslash \{i\}$, and let

$$\rho := \begin{cases} \mathbb{P}(T^+ \,|\, r)/\mathbb{P}(T \,|\, r) & \text{if } i \notin T, \\ \mathbb{P}(T \,|\, r)/\mathbb{P}(T^- \,|\, r) & \text{if } i \in T. \end{cases}$$

In other words, ρ is the conditional posterior probability ratio of including and not including t_i in the target set, given the observation result r and the remaining components of the target set.

The update rule is then: If $i \notin T$, add i to T with probability $\mathbb{P}(T^+ \,|\, r)/(\mathbb{P}(T^+ \,|\, r) + \mathbb{P}(T \,|\, r)) = \rho/(\rho + 1)$ (and leave T unchanged with the remaining probability $1/(\rho + 1)$). If $i \in T$, remove it with probability $1/(\rho + 1)$(and leave T unchanged with the remaining probability $\rho/(\rho + 1)$).

In this way, we cycle through all targets in either a fixed or random order in each round. This defines an ergodic Markov chain on T with the posterior as stationary distribution, from which we sample the quantities of interest during S sampling rounds after W warmup rounds to allow for the Markov chain to converge towards its stationary distribution.

We estimate the posterior marginals as follows. In round τ when updating target i, remember the value $p_i^{(\tau)} := \rho/(\rho + 1)$, where ρ is computed as described above. Then our estimate $\hat{\boldsymbol{\mu}}_i$ for μ_i is $\hat{\boldsymbol{\mu}}_i := \frac{1}{S} \sum_{\tau=W+1}^{W+S} p_i^{(\tau)}$, and our estimate for the target set size is $\hat{\boldsymbol{M}} := \sum_{i=1}^{m} \hat{\boldsymbol{\mu}}_i$.

Efficient computation of ρ-ratios. A key feature of this procedure is that the above ratios ρ can be efficiently computed in each step by taking advantage of the following observations.

Consider an update attempt $T \leftarrow T^+ = T \cup \{i\}$ with $i \notin T$, where, using Eqs. (1)–(3),

$$\begin{aligned} \rho &= \frac{\mathbb{P}(T^+)}{\mathbb{P}(T)} \cdot \frac{\mathbb{P}(r \,|\, T^+)}{\mathbb{P}(r \,|\, T)} \\ &= \frac{f_i}{1 - f_i} \cdot \prod_{j \in \mathcal{P}_i} \left(\frac{1-\eta_j(T^+)}{1-\eta_j(T)} \right)^{r_j} \cdot \left(\frac{\eta_j(T^+)}{\eta_j(T)} \right)^{1-r_j} \\ &= \frac{f_i}{1 - f_i} \cdot \prod_{j \in \mathcal{P}_i} \begin{cases} f_- & \text{if } r_j = 0, \\ \frac{1 - \eta_j(T) \cdot f_-}{1 - \eta_j(T)} & \text{if } r_j = 1 \end{cases} \\ &= \xi_i \cdot \prod_{\substack{j \in \mathcal{P}_i \\ r_j = 1}} \frac{1 - \eta_j(T) \cdot f_-}{1 - \eta_j(T)}, \end{aligned}$$

where $\xi_i := \frac{f_i}{1 - f_i} f_-^{|\{j \in \mathcal{P}_i : r_j = 0\}|}$. Note that in the prior ratio, everything except the i-th term cancels out, and in the likelihood ratio, all terms related to probes that do not hybridize to the i-th target also cancel out. The prior ratio and probability of necessarily false negative probes to include t_i in the target set is summarized in the factor ξ_i. Similarly, for an update attempt $T \leftarrow T \backslash \{i\}$, we have

$$\rho = \xi_i \cdot \prod_{\substack{j \in \mathcal{P}_i \\ r_j = 1}} \frac{1 - \eta_j(T)}{1 - \eta_j(T)/f_-}.$$

The ξ_i can be pre-computed and never change during the sampling phase, and the remaining product generally has few terms: the relevant probe

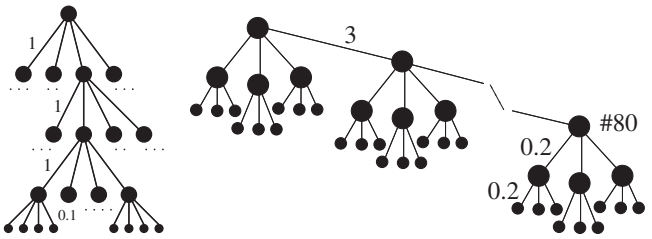


Fig. 2. *Left*: Balanced treemodel with 256 leaves. *Right*: Cherry tree model with 720 leaves.

Table 1. Summary statistics of the datasets. #Known refers to the number of known targets in the dataset, #Probes is the number of probe candidates selected by PROMIDE. #Hybs is the number of 1s in the hybridization matrix H^{basic}. The average number of hybridizations per probe and per target is shown in the next two columns. Finally, #Unknown denotes the number of unknown targets. Numbers are averages over the dataset instances

Name	Known	Probes	Hybs	H/probe	H/target	Unknown
`bal`	181	4038	10557	2.61	58.2	75
`cher`	539	8536	24485	2.87	45.4	181
`meio`	302	8837	16439	1.89	54.5	56

set $\mathcal{P}_i \cap \{j : r_j = 1\}$ can also be precomputed for every target i and will generally be sparse.

To evaluate the ratios within the products quickly, we maintain and update the vector $\eta = (\eta_j(T))_{j=1,\ldots,n} = \mathbb{P}(r_j = 0 \mid T)$ as defined in Eq. (1); in fact, we only require the elements η_j for which $r_j = 1$. Initially, T is empty, and $\eta_j(T) = 1 - f_+$ for all probes j. When T is enlarged to $T \cup \{i\}$ (resp. reduced to $T \backslash \{i\}$), we update $\eta_j \leftarrow \eta_j \cdot f_-$ (resp. $\eta_j \leftarrow \eta_j / f_-$) for all $j \in \mathcal{P}_i$ with $r_j = 1$.

3 EVALUATION

3.1 Datasets

We evaluate the proposed method on one biological dataset of organisms from the Meiobenthos and on two simulated datasets. Summary statistics of the datasets are shown in Table 1. The simulated datasets were generated with the REFORM (Random Evolutionary FORest Model) software [Rahmann, 2003b], freely available at http://gi.cebitec.uni-bielefeld.de/people/rahmann, that applies an evolutionary Markov process along a phylogenetic tree (specified in a small modeling language) to a random root sequence.

Simulated dataset `bal`. We generate 256 targets (leaf sequences) from a balanced tree as shown in Figure 1 (left). The tree has four levels below the root, and each internal node has out-degree four. For the internal branches, the evolutionary time is 1 percent of expected mutations (PEM), for the branches to the leaves, it is 0.1 PEM. Additionally, there are small insertion and deletion probabilities (details not shown). This leads to target sequence lengths between 970 and 1030, generated from a root sequence of length 1000. In order to have both known and unknown targets available, we traverse the tree top-down and prune the second and third child of each (internal or leaf) node we encounter with 20% probability. We generate 8 instances of this dataset with different random root sequences and random prunings. This leads to 146–210 known targets.

Simulated dataset `cher`. The tree consists of 80 nodes arranged in a linear chain with an inter-node distance of 3 PEM; see Figure 1 (right). Each chained node has three children in addition to the next node in the chain at distance 0.2 PEM, and each of these has in turn three children at the same distance. From the visual impression of this tree topology, we call this the cherry tree model. The 720 targets are generated from a root sequence of length 600, and their length ranges between 580 and 620. To generate unknown targets, the second child of each node is pruned away from the tree with 40% probability, leading to 527–555 known targets in the 8 generated instances of the dataset.

Real dataset `meio`. We use a set of 358 28S rDNA sequences from different organisms present in the Meiobenthos related by a phylogenetic tree [Markmann, 2000]. The set contains redundancies and many close homologs and finding unique probes is difficult [Schliep *et al.*, 2003, Kaderali and Schliep, 2002]. To generate unknown targets, we remove the the last leaf child of an internal node (if more than one exists) with 50% probability. We generated 5 instances of this dataset; in each distance, a different random target set is removed from the tree (cf. Table 1).

Probe selection. After randomly separating the sequences into known and unknown targets as described above, we use PROMIDE to select short oligonucleotide probes for the known targets. We pick all group-specific (groups were restricted to be of size 50 or below) 19–21-mers with Gibbs free energy between -20 and -19.5 kcal/mol at 40°C and a salt correction parameter of -2.6, according to the model parameters from [SantaLucia, 1998]. We create the extended hybridization matrix of all known targets against all probes, as described in Section 2.3.

We emphasize that the unknown targets have no influence on the probe selection process, but after the probes have been determined, we can of course compute their hybridization patterns. Although here we might face the problem of unclear signals (long common substrings), we take the approach that only exact full-length probe-target matches lead to a signal. The possibility of weaker cross-hybridization signals is handled by a correspondingly high false-positive error rate in our error model (up to $f_+ = 0.10$), see below.

3.2 Hybridization simulations and decoding

Simulations. We performed simulations of hybridization experiments to estimate the efficiency of our approach in detecting both known and unknown targets. We randomly sample target sets which are taken as the true result of the experiment. The sampling strategy is different for sets of known targets, for unknown targets, and mixed sets.

(1) `known`: We attempt to correctly detect the empty target set $T = \{\}$ and each of the $|\mathcal{T}|$ singleton sets $T = \{t_i\}$, $i = 1, \ldots, m$, where m varies for each dataset instance. For target sets cardinalities $2, \ldots, 6$, we sample 500 random sets each.

(2) `unknown`: For each unknown target (each removed leaf from the original phylogenetic tree), we determine its lowest existing ancestor in the remaining tree; this is an internal node. As discussed above, we take this node as a representative of any

Table 2. Average fraction of correctly identified true $|T|$ targets (hits) among the $|T|$ top randed targets given by the decoder for different datasets (rows) and different types of datasets (columns). For `unknown` and `mixed` datasets, a target is counted as a hit if either either the internal node representing the unknown target (colums ''Exact''), or taking a broader view, the node or its direct children (columns titled ''Fam.'' for family) are detected

Name	$f_+ = f_-$	`known`	`unknown` Extract	Fam.	`mixed` Exact	Fam.
bal	0.05	0.98	0.38	0.69	0.80	0.89
	0.1	0.94	0.36	0.68	0.77	0.86
cher	0.05	0.97	0.11	0.51	0.77	0.84
	0.1	0.94	0.11	0.54	0.71	0.83
meio	0.05	0.97	0.08	0.45	0.71	0.83
	0.1	0.96	0.06	0.44	0.70	0.82

unknown target in the subtree below it. Therefore, ideally, this node is the target that we would like to detect, although the hybridization pattern of the unknown target will generally differ from the ''majority vote'' pattern of the internal node. Also, different unknown targets may map to the same node. Because of these inherent difficulties, we only attempt to detect a single unknown target.

(3) `mixed`: Finally, we attempt mixed sets with exactly one unknown target and between 1 and 3 known targets. For each cardinality, 500 random sets are sampled.

For each target set $T \subset \{1, \ldots, m + |\mathcal{I}|\}$, we simulate 10 independent hybridization results according to the error model described in Section 2.4, i.e., for each probe p_j, we determine the number of targets in T to which p_j would hybridize and let each hybridization fail independently with probability f_-; finally, there is a probability of f_+ that p_j shows an unspecific positive signal. This simulation was performed once with error rates $f_+ = f_- = 0.05$ and again with $f_+ = f_- = 0.1$.

Decoding. We ran our own `TPDC` decoding software with a uniform prior $f = (f_i)$, $i = 1, \ldots, m + |\mathcal{I}|$ on all targets such that $\sum_i f_i = 3$. The error parameters $f_- = f_+ \in \{0.05, 0.1\}$ were the same as used in the simulations. In practice, the error rates are not known and must be estimated. After 200 warmup rounds, the marginal target posteriors were estimated from the subsequent 2000 rounds; these values were found to be sufficiently accurate when compared to substantially longer runs. The output consists of a list of targets sorted by marginal posterior and additional diagnostics. Only targets with a posterior exceeding 0.001 were included in further analysis.

3.3 Results

Ideally, we observe exactly the true targets as the top entries of the list returned by the decoder. Depending on the similarity of hybridization patterns and on the noise level, we must expect a number of high-posterior targets that do not belong to the target set. In some of those cases, the ''offender'' is likely to be a close relative of the true target. We take this fact into account in our evaluation.

The success rates of our approach for a total of about 2,292,380 simulated experiments are summarized in Table 2. Simulation results for the datasets bal, cher are averaged over the 8 instances, for meio over the 5 instances, over all target set cardinalities, and over the 10 repetitions of simulated hybridizations.

Our method is able to correctly identify over 94% of known targets in simulated experiments with realistic error rates. If there are neither known nor unknown targets present, the maximal target posterior observed in all repetitions and data sets was 0.15 and posteriors exceeding the 0.001 posterior threshold were predominantly (over 95%) below 0.01, implying a negligible false positive rate. The results for unknown targets suggest that our simple approach for defining the hybridization pattern of its parent is not sufficient. There is a jump in performance when also direct children are counted as a hit. Then, up to 70% of the unknowns were correctly assigned to their clade in the complete tree. Detailed summaries for the 2,292,380 simulated experiments are found in the supplementary material.

4 DISCUSSION

We present an approach for decoding hybridizations experiments when targets are related by a phylogenetic tree and non-unique oligonucleotide probes are used in a statistical group testing setting. Hybridization patterns of internal nodes of the tree are obtained from leaves based on a majority rule as typical patterns for unknown targets in the respective subtree. A Bayesian framework combined with a Markov chain Monte Carlo approach allows efficient and robust estimation of target posterior marginals.

Our method correctly identifies over 94% of known targets, and about 45% to 70% of unknown targets were correctly assigned to their clades in the phylogenetic tree. The lower figures for unknown targets are explained by the fact that the majority-vote hybridization patterns of the internal nodes do not (and cannot) match exactly the hybridization patterns of unknown targets.

We found that our estimate of the target set size $|T|$ matches the true value in virtually all of the cases when rounded to the nearest integer. It follows that the rate of falsely identified targets is between 2% and 6% for known targets.

More detailed analysis of the high-ranking targets may improve the resolution of the method in the presence of unknowns, as we correctly identify clades but do not provide a statistical test for the hypothesis that unknowns belonging to this clade are present.

In a practical application of the method, the true target set size $|T|$ and the error rates f_+, f_- for the decoding procedure will be unknown. However, we can estimate $|T|$ by the sum of the posterior marginals, and our results show that the method is robust, even for relatively high error rates, which makes it reasonable to use with slight overestimates of error-rates, possibly at the expense of less pronounced posterior magnitudes. For the robustness of the method, a high probe coverage per target is necessary, and future work may show to which degree the probe set may be reduced without affecting our ability to detect unknown targets too severely.

Our results on biological and simulated data demonstrate that we can cope effectively with the incomplete phylogenies available in practical applications and that the method is robust with respect to evolution of targets between time of design and time of experiment. We are not aware of previous studies that consider the problem of recognizing unknown or fast evolving targets in such a manner.

ACKNOWLEDGEMENTS

Many thanks to Diethard Tautz and Win Hide for helpful discussions. We also thank Jonas Heise for help with preprocessing the Meiobenthos dataset.

REFERENCES

L. Kaderali and A. Schliep. Selecting signature oligonucleotides to identify organisms using DNA arrays. *Bioinformatics*, 18(10):1340–1349, Oct 2002.

G. W. Klau, S. Rahmann, A. Schliep, M. Vingron, and K. Reinert. Optimal robust non-unique probe selection using Integer Linear Programming. *Bioinformatics*, 20(Suppl 1):i186–i193, Aug 2004.

M. Markmann. *Entwicklung und Anwendung einer 28S rDNA-Sequenzdatenbank zur Aufschlüsselung der Artenvielfalt limnischer Meiobenthosfauna im Hinblick auf den Einsatz moderner Chiptechnologie*. PhD thesis, University of Munich, 2000.

C. Putonti, S. Chumakov, R. Mitra, G. E. Fox, R. C. Willson, and Y. Fofanov. Human-blind probes and primers for dengue virus identification. *FEBS J*, 273(2):398–408, Jan 2006.

S. Rahmann. Rapid large-scale oligonucleotide selection for microarrays. In *Proceedings of the First IEEE Computer Society Bioinformatics Conference (CSB)*, pages 54–63. IEEE, 2002.

S. Rahmann. Fast large scale oligonucleotide selection using the longest common factor approach. *Journal of Bioinformatics and Computational Biology*, 1(2):343–361, 2003a.

S. Rahmann. REFORM (Random Evolutionary FORests Modeling software), 2003b.

S. Rash and D. Gusfield. String barcoding: Uncovering optimal virus signatures. In *Proceedings of RECOMB 2002*, pages 254–261, April 2002.

J. SantaLucia. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences of the U.S.A.*, 95:1460–1465, 1998.

A. Schliep, D. C. Torney, and S. Rahmann. Group testing with DNA chips: Generating designs and decoding experiments. In *Proceedings of the 2nd IEEE Computer Society Bioinformatics Conference (CSB 2003)*, pages 84–93. IEEE, 2003.

D. Wang, A. Urisman, Y.-T. Liu, M. Springer, T. G. Ksiazek, D. D. Erdman, E. R. Mardis, M. Hickenbotham, V. Magrini, J. Eldred, J. P. Latreille, R. K. Wilson, D. Ganem, and J. L. DeRisi. Viral discovery and sequence recovery using DNA microarrays. *PLoS Biol*, 1(2):E2, Nov 2003.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e431–e439
doi:10.1093/bioinformatics/btl238

Integrating copy number polymorphisms into array CGH analysis using a robust HMM

Sohrab P. Shah[1,*], Xiang Xuan[1], Ron J. DeLeeuw[2], Mehrnoush Khojasteh[2], Wan L. Lam[2], Raymond Ng[1] and Kevin P. Murphy[1]

[1]Department of Computer Science, University of British Columbia, 201-2366 Main Mall Vancouver, BC V6T 1Z4 Canada and [2]British Columbia Cancer Research Centre, 675 West 10th Avenue Vancouver, BC V5Z 1L3 Canada

ABSTRACT

Motivation: Array comparative genomic hybridization (aCGH) is a pervasive technique used to identify chromosomal aberrations in human diseases, including cancer. Aberrations are defined as regions of increased or decreased DNA copy number, relative to a normal sample. Accurately identifying the locations of these aberrations has many important medical applications. Unfortunately, the observed copy number changes are often corrupted by various sources of noise, making the boundaries hard to detect. One popular current technique uses hidden Markov models (HMMs) to divide the signal into regions of constant copy number called segments; a subsequent classification phase labels each segment as a gain, a loss or neutral. Unfortunately, standard HMMs are sensitive to outliers, causing over-segmentation, where segments erroneously span very short regions.

Results: We propose a simple modification that makes the HMM robust to such outliers. More importantly, this modification allows us to exploit prior knowledge about the likely location of "outliers", which are often due to copy number polymorphisms (CNPs). By "explaining away" these outliers with prior knowledge about the locations of CNPs, we can focus attention on the more clinically relevant aberrated regions. We show significant improvements over the current state of the art technique (DNAcopy with MergeLevels) on previously published data from mantle cell lymphoma cell lines, and on published benchmark synthetic data augmented with outliers.

Availability: Source code written in Matlab is available from http://www.cs.ubc.ca/~sshah/acgh.

Contact: sshah@cs.ubc.ca

1 INTRODUCTION

Array comparative genomic hybridization (aCGH) is a high-throughput genetic technique to measure DNA copy number changes in a disease sample compared to a normal sample [20]. Chromosomal aberrations that exhibit DNA copy number changes are indicative of numerous diseases including cancer and mental retardation. Identifying such aberrations can help to locate diagnostically important regions in the genome, that harbour important genes. For example, oncogenes or tumour suppressor genes contained in aberrated regions could in turn exhibit differential expression due to the copy number changes in the DNA. Application of aCGH is widespread in molecular analysis of cancer and holds great promise as a technique to identify clinically relevant diagnostic biomarkers.

The aCGH technique is based on spotting clones that span a discrete region in the human genome on an array. The size and number of clones vary depending on the technological platform and the desired resolution: see Pinkel and Albertson [20] for a review. In this paper, we use aCGH data from eight mantle cell lymphoma (MCL) cell lines (see deLeeuw *et al*. [4] for details) generated using sub-megabase resolution tiling arrays (SMRT) [13]. We use the midpoint of the clone along the chromosome to denote its location. The output of all aCGH platforms is represented as a $\log_2$ ratio of the reference and tumour fluorescence intensities for every clone in the array. The $\log_2$ ratios are expected to be proportional to copy numbers. In a neutral state, one would expect to see $\log_2(2/2) = 0$; with one copy lost, one would expect to see $\log_2(1/2) = -1$; with one gain $\log_2(3/2) = 1.58$, etc. The goal of analysis techniques is to detect contiguous regions, that are expected to share the same mean $\log_2$ ratio. We call these regions segments. The identification of segments is called ''segmentation''. Once segments are identified, each segment is labeled as a loss, neutral or gain (sometimes it is useful to distinguish gains of 1 copy from gains of more than 1). This latter task is called ''classification''.

In reality, segmentation and classification of the data are much more difficult than the above description suggests. Figure 3 (A) shows a typical plot of aCGH data for chromosome 1 from MCL cell line HBL-2 (see Section 3.1 for more details on MCL). The yellow squares represent clones that are found in a region of loss identified manually by an expert [4]. Similarly, blue circles represent clones in a region of gain. The figure demonstrates that although copy number changes in DNA is a theoretically discrete process, the intensity ratios for aCGH do not produce a clean piecewise constant signal. Also note that aberrated regions tend to span contiguous sets of clones along a chromosome. This suggests that any analysis technique should exploit such spatial correlation.

In Figure 3 (A), we also depict 'outlying' clones (detected by eye) with light blue stars. Treating such points as inliers can significantly affect the remaining points, by causing over-segmentation, resulting

*To whom correspondence should be addressed.

in segments that span only a single clone, for example. There are several possible causes of such outliers. The first is some kind of measurement noise, or mislabeling (sometimes the locations of clones is mis-recorded). Second, there is the possibility that the single clone outliers correspond to known locations of copy number polymorphisms (CNPs). Finally, they could truly represent aberrated regions. In our experience, this is rare.

The full impact of CNPs on aCGH analysis is not yet known, however indications from two recent large scale studies by Sebat *et al*. [22] and Iafrate *et al*. [12] measuring background frequencies of copy number variations in the normal human population have revealed hundreds of loci in the genome that are polymorphic in copy number. Buckley *et al*. [2] suggest that the results produced by these two studies represent the ''tip of the CNP iceberg''. For example Sebat *et al*. report a CNP at a gene involved in food intake, suggesting a differential propensity for obesity. They also report CNPs at loci related to neurological development and at loci implicated in leukemia and breast cancer drug resistance [22]. These latter examples indicate that for cancer studies, the 'baseline' copy number should be considered when assessing aberrations. We anticipate that the impact of CNPs will be greater on high-resolution arrays and/or full genome coverage arrays, as they are intended to reveal all aberrations in a sample and will detect a larger number of CNPs. Note that the MCL data is both high-resolution and full coverage and therefore is likely to contain CNPs.

1.1 Our contributions

In this paper, we introduce a joint classification and segmentation method that is designed to handle outliers and integrate CNP knowledge into the analysis. Our method extends the standard HMM framework, outlined in Scott [21] and proposed for aCGH in Guha *et al*. [9]. The basic idea is to replace the Gaussian observation model with a mixture of Gaussians; one mixture component represents the $\log_2$ ratio we would expect from the given state (loss, neutral or gain); the other mixture component represents the $\log_2$ ratio we would expect from an outlier. This simple change makes the model much more robust.

More significantly, we can incorporate knowledge about CNPs into the mixing weights of the mixture model. That is, we can set the prior probability of using the outlier component at location i to the known frequency of CNPs at location i, if i overlaps with a known CNP location; otherwise we set it to the general background outlier probability (which is estimated from data). We explain our model in more detail in Section 2.1.

Several authors (e.g., [9,21]) propose estimating the parameters of the HMM using MCMC (Markov chain Monte Carlo) techniques, as opposed to the more common EM (expectation maximization) algorithm. The advantage of MCMC is that it provides full posterior estimates over the parameters, rather than just point estimates, thus properly modeling uncertainty (see e.g., [8] for an introduction to MCMC and Bayesian data modeling). MCMC also partly mitigates problems with local minima that EM is well known to suffer from. It also turns out to be simpler to exploit informative prior constraints in a sampling framework than in an optimization framework. We explain how to perform efficient MCMC in Section 2.4.

We first evaluate performance of our model on real data representing aCGH profiles from eight MCL cell lines published in deLeeuw *et al*. [4] in a study aimed at identifying important signature regions in non-Hodgkin's lymphoma. This data set contains ground truth annotation of regions of gain and loss, some of which are recurrent across cell lines. In addition some of these aberrations have been validated in the laboratory. Using this rich data set, we were able to assess performance quantitatively using standard performance metrics. We compare our method to DNAcopy+MergeLevels (using default parameters), which has been shown in two previous comparative studies [24,16] to be a leading current method. Henceforth we will refer to this method as MergeLevels. Having established that our method is better than current techniques, we then validate our findings on an additional synthetic data set, which we believe to be 'harder' than the real data. The advantages of using synthetic data are two-fold. First, the ground truth locations of the aberrations are known. Second, we can control the difficulty of the problem. We used data published in Willenbrock and Fridlyand [24]. This data is considerably harder (but more realistic) than other synthetic datasets used in earlier papers. We make the Willenbrock and Fridlyand data even harder by adding outliers, to check the robustness of our method and to validate results obtained using MCL data. Our results are in Section 3, which we discuss in Section 4.

1.2 Related work

A recent survey paper by Lai *et al*. [16] describes and evaluates eleven algorithms for aCGH data analysis. We can loosely group these methods into three main approaches: smoothing, segmentation, and combined segmentation and classification. Smoothing approaches such as Quantreg, developed by Eilers and Menezes [5], and the wavelet approach of Hsu *et al*. [10], attempt to fit a curve to the data, while handling abrupt changes. Smoothing methods generally filter the data using a fixed size window, and therefore will be unable to detect outliers or CNPs that span a single clone. In addition, they are primarily designed as a visual aid to interpret the data and do not accomplish the main objective of automatically identifying aberrated clones.

As mentioned previously, segmentation methods identify contiguous sets of clones (segments) that share the same mean $\log_2$ ratio. The output of the segmentation methods usually consists of the boundaries and means of the segments. The clones within a segment are assumed to share the same copy number. We refer to the boundaries of segments as breakpoints. Examples of segmentation algorithms include DNACopy [18], which is based on a recursive circular binary segmentation algorithm; CGHSeg [19] which uses a penalised likelihood model to determine breakpoints; aCGH-Smooth [14], which uses a genetic algorithm to find breakpoints; and the GLAD method of Hupe *et al*. [11], which includes a median absolute deviation model to explicitly treat outliers as separate from its surrounding segment. In Lai's comparison, CGHSeg and DNACopy are consistently the best. Willenbrock and Fridlyand [24] compared performance of DNACopy and GLAD and report better performance with DNACopy. We therefore use DNAcopy as our baseline model. Note that Lai *et al*. [16] determined that as the noise level in the data increases, all segmentation methods—including DNAcopy, show less than satisfactory results.

A general limitation of segmentation is that the output needs to be further analysed in order to infer which segments are aberrated regions, i.e., to ''call'' the gains and losses. Methods such as GLADMerge [11] and MergeLevels [24] perform this post-processing task by merging together segments with ''similar'' mean levels, and then classifying them. However, as noted by

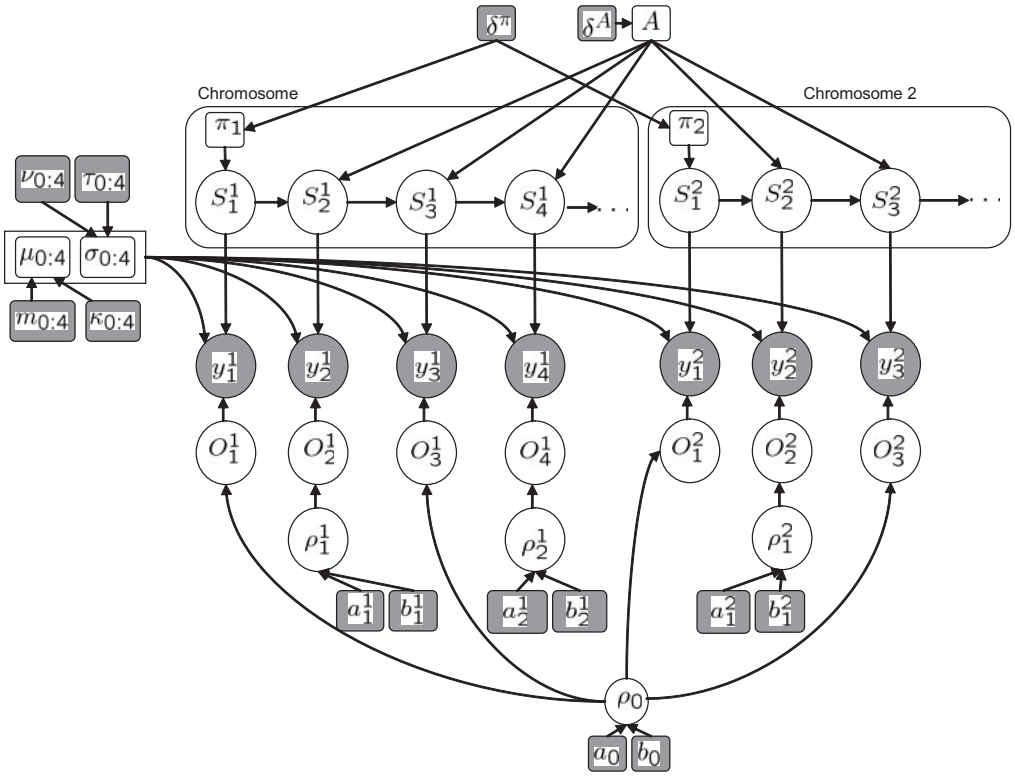


Fig. 1. Our model represented as a Bayesian network. Square nodes are parameters, round nodes are random variables. Shaded nodes are observed (known), unshaded nodes are hidden (unknown). We depict the start of 2 chromosomes (indicated by large rounded rectangles). Let c denote the chromosome, i represent the location along the chromosome and k represent the kth LSP on the chromosome; π_c is the initial state distribution of chromosome c; δ^π are hyperparameters for the π_c's; S_i^c is the state; y_i^c is the observation ($\log_2$ ratio); O_i^c indicates if this is an outlier or not; $\mu_{1:4}$ and $\sigma_{1:4}$ are the means and variances of states 1 to 4; μ_0 and σ_0 is the mean and variance of the outlier state; ρ_k^c is the probability of outlier for LSP locations; ρ_0 is the general background outlier probability; A is the Markov chain transition matrix; δ are the hyperparameters for A. For state j, m_j, τ_j are hyperparameters for μ_j; α_j, β_j are hyperparameters for σ_j; a_k^c, b_k^c are hyperparameters for ρ_k^c and are determined by LSPs; Hyper-parameters are shown shaded since they must be set by the user. In this example, we have assumed that locations 2 and 4 on chromosome 1 and location 2 on chromosome 2 correspond to known CNPs; other locations use the background outlier probability ρ_0. Hence the prior on O_1^1, O_3^1, O_1^2 and O_3^2 are all the same and equal to ρ_0.

Engler *et al*. [6] and Willenbrock and Fridlyand [24], it is much better to perform the segmentation and classification simultaneously, since the class labels can help with the segmentation as well as vice versa.

An obvious way to perform simultaneous segmentation and classification is to use an HMM. The first approach to do this was by Fridyland *et al*. [7]. However, in their approach, the states of the HMM do not have any intrinsic meaning (they are just indices to represent a discrete number of mean levels, typically $K = 5$). Hence post-processing was necessary to determine the labels. Guha *et al*. [9] modify this to use a ''supervised'' 4-state HMM, where the states are defined to mean loss, neutral, one-gain or multiple-gain. The advantage of this is two-fold: first, it is easy to perform simultaneous segmentation *and classification* using the Viterbi algorithm; secondly, we can impose informative priors on the parameters, since they now have biological meaning. This paper extends the model by adding robustness to outliers and location-specific priors (LSPs), which can be used to encode CNPs.

In addition to the work mentioned above, two recent papers have explored some interesting variations. Broet and Richardson [1] propose using a latent 1D Gaussian random field, as opposed to a latent 1D discrete random field (i.e., an HMM), to model spatial correlation between levels. However, this does not solve the classification problem. Engler *et al*. [6] introduce spatial dependence by breaking the data into overlapping triples, and then using a hierarchical random effects model. Unfortunately, because the triples are overlapping, the data is over-counted, so optimizing the likelihood turns out to be intractable. Instead, they compute a local maximum of the pseudolikelihood. We also use a hierarchical Bayesian model, but we are able to compute posterior estimates using an exact likelihood function.

2 METHODS

2.1 A mixture model HMM that considers outliers

Our model, sketched in Figure 1, is similar to the 4-state HMM in Guha *et al*. [9], where the states represent loss, neutral, one-gain and multiple-gain. (We also tried a 3-state model, where we combined the gain states, but results were not as good.) The main difference from Guha is that in our model the observation density is a mixture of 2 Gaussians, one representing inlier (clones belonging to one of the states) and the other representing outlier. We introduce binary indicator variables $O_i \in \{0, 1\}$ where $O_i = 1$ means

Table 1. User settable hyper parameters for our model, along with the values we used for the Willenbrock synthetic data and the real MCL data. f_i is the frequency of a known CNP at position i. In the synthetic data, we set this to 0.001. To help interpret these numbers, recall that the mean of a *beta*(a,b) random variable is $a/(a+b)$, and the mean of a $Ga(\alpha,\beta)$ random variable is α/β. In particular, this means $\sigma_1^2 = 0.1$ for synthetic and 0.07 for MCL, $\sigma_2^2 = 0.01$ for synthetic and 0.05 for MCL, etc

Parameter	Description	Synthetic	MCL
δ	Dirichlet prior on transition matrix A	1, 1, 1, 1	1, 1, 1, 1
$\alpha_{1:4}$	shape of gamma prior on inverse variances σ^{-2}	10, 100, 5, 5	15, 20, 10, 10
$\beta_{1:4}$	scale of gamma prior on inverse variances σ^{-2}	1, 1, 1, 1	1, 1, 1, 1
$m_{1:4}$	prior mean on means μ	−0.1, 0, 0.58, 1	−0.4, 0, 0.3, 0.5
$\tau_{1:4}$	prior variance on means μ	0.5, 0.001, 1, 1	0.2, 0.1, 0.2, 0.2
a_0	a for beta prior for ρ_0 (prob of outlier)	0.01	0.00001
b_0	b for beta prior for ρ_0 (prob of outlier)	0.99	0.99999
a_i	a for beta prior for ρ_i (prob of outlier at CNP i)	0.001	f_i
b_i	b for beta prior for ρ_i (prob of outlier at CNP i)	0.999	$1 - f_i$

location i is an outlier, and $O_i = 0$ means it is an inlier. We model the outlier distribution with a Gaussian, μ_0, σ_0^2.

Using the mixture of Gaussians, the class-conditional density becomes

$$p(y_i \mid O_i, S_i = s) = \begin{cases} Gauss(y_i \mid \mu_0, \sigma_0) & \text{if } O_i = 1 \\ Gauss(y_i \mid \mu_s, \sigma_s) & \text{if } O_i = 0 \end{cases} \quad (1)$$

where y_i is the $\log_2$ ratio for clone i where the clones are ordered by their physical location on a chromosome. $S_i = s$ is the state label at i, where s is a discrete random variable $\in\{1,2,3,4\}$ with 1 corresponding to the loss state; 2 to the neutral state; 3 to the one-gain state; and 4 to the multiple-gain state. The unobserved sequence of states is governed by Markovian dynamics encoded in the transition matrix A. The transition matrix therefore models the spatial correlation expected to occur in the data. The clones labeled with a given state s are generated from a a common Gaussian distribution with μ_s and σ_s. The initial state distribution for $i = 1$ is multinomial random variable π that models the probability of starting in each state. In the observation density of our model, O_i acts like a ''switching parent'' variable, which selects between the outlier parameters μ_0, σ_0 or the inlier parameters, μ_s, σ_s. The O_i variables are modeled as conditionally independent. Hence there are no Markovian dynamics on the outliers. This allows the model to make temporary ''excursions'' to the outlier state, without incurring any ''penalty'' implicitly encoded by the state transition matrix.

Modeling the outliers as conditionally independent also allows us to encode CNPs. For each location that is known to be a CNP, we have an outlier probability, $\rho_i = p(O_i = 1)$; for all other locations, we have the ''background'' outlier probability, ρ_0.

2.2 Parameter estimation using 'pooling' across chromosomes

In addition to the outlier extension, we extend our model by estimating some of the parameters using pooled data across all chromosomes in the sample. Parameters A, μ, σ, ρ_0 are estimated by pooling. This assumes that the posterior distributions of these parameters are expected to be consistent across chromosomes, and therefore pooling is advantageous as their estimates are guided by more data. Moreover, the algorithm is more likely to 'visit' all the states by pooling the data, resulting in more robust estimates of the mean and variance of each state. However, not all the parameters can be estimated in this way. Sampling of the states S must be estimated on each chromosome separately as there is no physical interpretation for a state dependency between location 1 on chromosome c and the terminal location on chromosome $c - 1$. π must also be estimated independently for each chromosome since the telomeric regions of the chromosomes can have gains, losses or remain neutral and these initial states are not expected to be consistent across chromosomes. The model, showing pooling across chromosomes and the outlier parameters, is depicted as a Bayesian network in Figure 1. Since the figure shows pooling across chromosomes, (S,y,O,π and ρ) are indexed by both chromosome and location. The chromosome index was omitted above for notational clarity.

An obvious extension of pooling data across chromosomes is to pool data across samples, such as in Engler *et al*. [6]. However, due to numerous factors that are sample-specific such as ploidy of the tumour genome and proportion of tumour cells in the sample[24], we do not assume that mean levels of copy number change will be consistent across samples. Therefore we do not estimate mean levels of the states across samples. However, we suggest that jointly considering samples has considerable value for goals other than classification. Indeed, multiple aCGH samples of the same cancer subtype have something in common—this is precisely what scientists hope to discover! Presumably, multiple samples of the same cancer subtype will exhibit commonalities such as minimally overlapping aberrations (see [4] for examples in MCL cell lines) and locations of breakpoints. Detecting such features is the subject of future work, and for now, we limit our attention to modeling samples separately.

2.3 Priors

We use standard conjugate priors (see e.g., [8]) for all the parameters, as follows:

$$p(\mu_s \mid \sigma_s) = Gauss(m_s, \tau_s \sigma_s) \quad (2)$$

$$p(\sigma_s^{-2}) = Ga(\alpha_s, \beta_s) \quad (3)$$

$$p(A) = Dir(\delta) \quad (4)$$

$$p(\rho_i) = Beta(a_i, b_i) \quad (5)$$

As is apparent, these priors themselves have parameters, called hyper-parameters. We set these by hand. Specifically, we use a small fraction of validation data in order to estimate (by eye) the typical mean and variance of the loss, neutral and gain states. A more rigorous Bayesian approach would be to extend the hierarchy even further, and add priors to the hyperparameters. Previous work has shown that 3 levels of hierarchy (parameters, hyper-parameters, and hyper-hyper-parameters) is usually sufficient to obtain robustness to (hyper-hyper-)parameter settings. We plan to investigate this in the future. A summary of all the user-settable parameters is shown in Table 1.

Prior knowledge about CNPs is encoded as follows. Locations $i \in P$ which are known to come from CNPs get an adjustable parameter ρ_i which reflects the probability of outlier at that location. The parameters of the (*Beta*) prior on ρ_i is set so that the expected value of ρ_i is equal to the frequency of polymorphisms at that location in the population. Locations $i \notin P$, which are

initialize parameters to prior mean
initialize $o^1_{1:n}$ sensibly (eg set $O_i = 1$ if obviously an outlier)
for each iteration t
  Compute local evidence $B^t_i(s) = p(y_i|S_i = s, o^t_i, \mu^t, \sigma^t)$ using Equation 1
  Block B_1: for each chromosome c:
    sample $s^{t+1}_{c_1:c_n}|y_{c_1:c_n}, A^t, B^t, \pi^t_c$ with forwards-backwards
    sample $\pi^{t+1}_c|s^{t+1}_{c_1}$
  Block B_2: sample $o^{t+1}_{1:n}|y, s^{t+1}, \rho^t, \mu^t, \sigma^t$ independently
  Block B_3: sample $\mu^{t+1}_{0:4}|\sigma^t, y, s^{t+1}_{1:n}, o^{t+1}_{1:n}$
  Block B_4: sample $\sigma^{t+1}_{0:4}|y, s^{t+1}_{1:n}, o^{t+1}_{1:n}$
  Block B_5: sample $\rho^{t+1}_{0:C}|o^{t+1}_{1:n}$ independently
  Block B_6: sample $A^{t+1}|S^{t+1}_{1:n}$
next t

Fig. 2. Pseudo code for the *pooled* algorithm. c_1 and c_n indicate the initial and terminal positions on chromosome c. n indicates the total number of clones in the sample.

not known to come from CNPs, share the same parameter ρ_0, which represents the background probability of outlier. The (*Beta*) prior on ρ_0 is set so that the expected value of ρ_0 is equal to the expected fraction of outliers, which we estimate by eye on a per dataset basis (once for synthetic data, once for MCL). We will let $C = |P|$ be the number of CNP locations, so ρ is a vector of length $C + 1$.

In order to ensure the model is identifiable (i.e., to avoid label switching), we enforce the following constraint on the mean parameters: $\mu_1 < \mu_2 < \mu_3 < \mu_4$, where the states represent loss, neutral, one-gain and multiple-gain. (We do this using a truncated Gaussian prior.) We impose a similar constraint on the state variances: $\sigma_{3,4} \geq \sigma_1 \geq \sigma_2$, which means that the gain states have higher variance than the loss state, which has higher variance than the neutral state (an empirical fact about most aCGH data). (We encode this using a truncated Gamma prior.) Note that handling truncated priors in EM (for MAP estimation) is harder than with MCMC, since it would require constrained optimization methods in the M step. Indeed, EM is usually only used to fit unidentifiable HMMs.

2.4 Algorithm

The algorithm is sketched in Figure 2. The output of the algorithm is the following: estimates of the states $\gamma_i(s) = p(S_i = s \,|\, y_{1:n})$ and outlier probabilities $\omega_i(o) = p(O_i = o \,|\, y_{1:n})$, as well as estimates of the parameters, $p(\theta \,|\, y_{1:n})$ where $\theta = (\pi, A, \mu, \pm\sigma, \rho)$. We use an MCMC algorithm called block Gibbs sampling to infer these quantities. The key to making this efficient is to use the forwards-filtering backwards-sampling algorithm for HMMs [21]. This is very similar to the more familiar forwards-backwards and Viterbi algorithms, except we sample state sequences from their posterior, rather than computing the most probable sequence or marginal state probabilities. Conditioned on knowing the states, it is easy to update the parameters of the model. The same intuition is used in EM, but the advantage of sampling is that we can model uncertainty in the parameters more easily.

To evaluate the effect of pooling across chromosomes, we implemented our model in two modes, one which models each chromosome of each sample independently (*single*), and the other which estimates the parameters of the model by summarizing over chromosomes in each sample (*pooled*), as shown in Figure 1. The algorithm for *pooled* mode is shown in Figure 2. To reduce the *pooled* model to the *single* model, we consider a single chromosome at a time and assign each chromosome its own set of private parameters A, μ, σ and ρ_0. We assess the relative performance of these two implementations in Section 3.

The running time is $O(NT)$ where N is the number of clones in the input and $T \sim 100$ is the number of MCMC iterations needed to obtain convergence (which we assess informally by monitoring quantities of interest by eye). The method is entirely standard except for the update of ρ. We update the ρ_i parameters (based on the sampled value of O_i) for those locations $i \in P$ known to correspond to CNPs; for all other locations, we update ρ_0 using the sufficient statistic $\sum_{i \notin P} O_i$. In *pooled* mode, the forward-filtering, backwards-sampling step (which samples a state sequence) is performed on each chromosome separately, as imposing dependencies on the terminal clone from one chromosome and the initial clone of the next chromosome is non-sensical. In both algorithm variants, we parameterize π separately with its own Dirichlet distribution; this allows the use of Gibbs sampling to update the hyperparameters by simple counting. In contrast, Guha *s*olve for π using the stationary distribution of A, which requires a Metropolis-Hastings step [9]. Currently we estimate $\hat{\gamma}_i(s) = p(S_i = s \,|\, y_{1:n})$ by counting the number of inlier samples for which $S_i = s$.

2.5 Evaluation methods

We evaluated our algorithm by calculating precision and recall for aberrations (gains and losses grouped together). Given a ground truth labeling and a predicted labeling of the clones (obtained by taking the max $p(S_i \,|\, y)$ probability), let *ntp* be the number of true positives (correctly predicted aberrations), let *nt* be the number of true aberrations, and let *np* be the number of predicted aberrations. Recall is defined as $\frac{ntp}{nt}$, meaning the proportion of true aberrations detected by the algorithm. Precision is defined as $\frac{ntp}{np}$, meaning the proportion of predicted aberrations that are true. By varying the threshold on the probabilities, we can vary the trade-off between precision and recall. To summarize the precision-recall curve in one number, we use the F-measure, which is the geometric mean:

$$F = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

To summarize accuracy results over many samples or chromosomes, we use distributions of F-measures.

We now explain how we modify the above method to handle outliers. We first compute the posterior probability of outlier for each clone, $p(O_i = 1 \,|\, y)$. We then rank these probabilities and take the top p_o% of them; finally, we select those whose absolute probability is above a threshold t_o. We then remove all those clones, which are deemed outliers, and compute precision-recall on the remaining locations in the usual way. Note that these parameters are not part of the algorithm. They are only used in the evaluation process. We use $p_o = 10\%$ and $t_o = 0.01$.

3 RESULTS

To systematically test our approach, we ran three variants of our algorithm on each data set:

- The baseline HMM (Base-HMM) which clamps the probability of outlier at each location to 0, $p(O_i = 1) = 0.0$. This reduces the model to an HMM with no outlier processing ability, as in [9].
- The robust HMM (Rob-HMM), which uses $C = 0$ CNPs but updates the global outlier probability $p(\rho_0 \,|\, y)$ given data from all locations.
- The robust HMM augmented with location specific prior (LSP) knowledge (LSP-HMM). In particular, we allow all locations $i \in P$ to have their own prior probability of outlier, ρ_i.

For each of these variants, we also ran the algorithm in *single* and *pooled* mode. We also ran MergeLevels, considered to be the current best method.

3.1 Mantle cell lymphoma cell line data

To illustrate the performance of our method on real data, we used a set of 8 MCL cell lines (Granta-519, HBL-2, NCEB-1, Rec-1, SP49, UPN-1, Z138C and JVM-2) whose aCGH profiles were manually

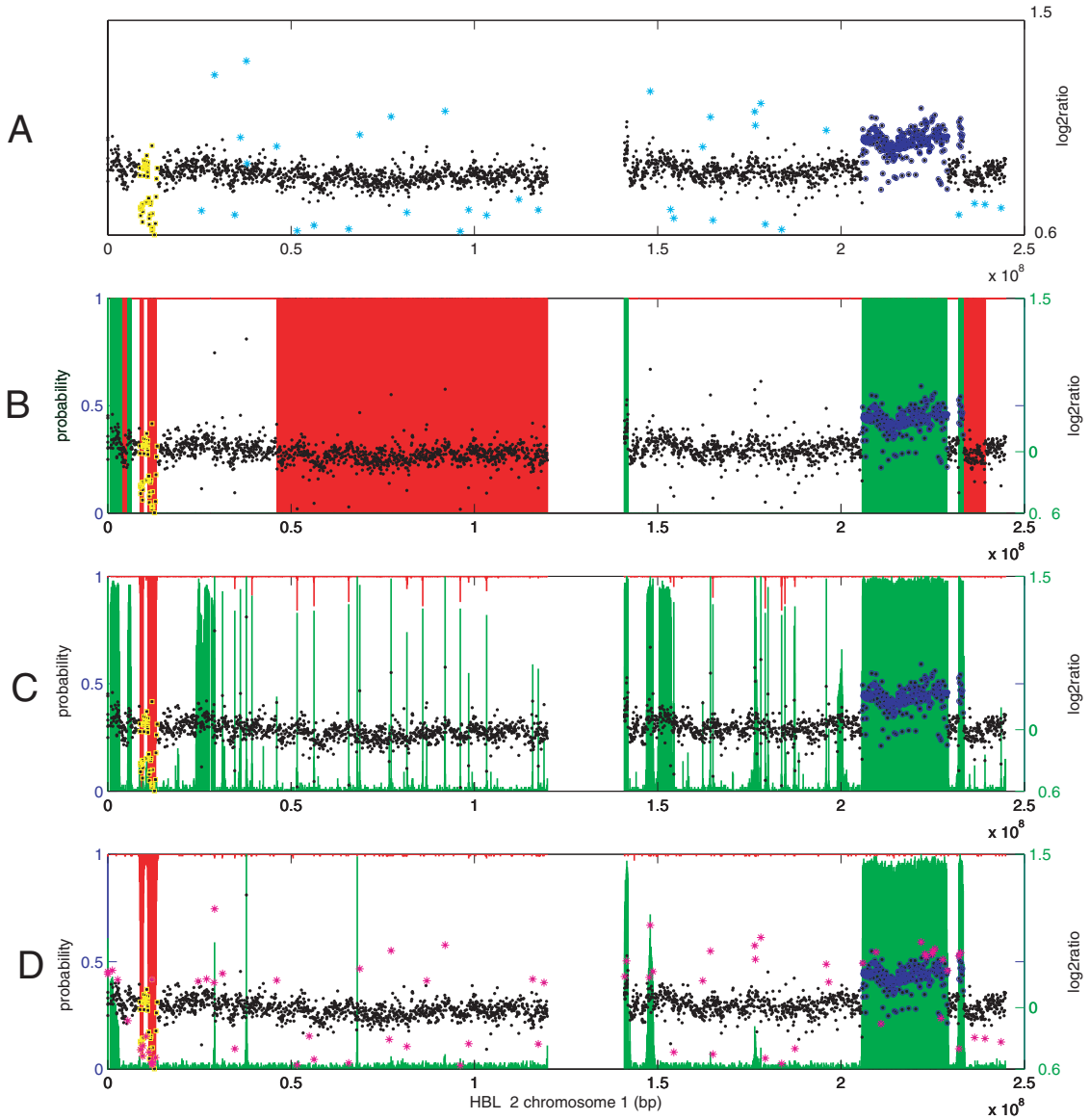


Fig. 3. Array CGH profile for chromosome 1 of the MCL cell line HBL2. The x axis for all panels indicate position in nucleotides (bp) along the chromosome. Panel **A** shows the $\log_2$ ratios (right axis) plotted against the position of each clone on the chromosome. The yellow squares indicate clones contained in a region labeled as a loss by an expert. The blue circles similarly indicate clones that are in a gain region. Clones marked with light blue stars indicate outliers. Panel **B** shows the predicted gains (vertical green bars) and losses (vertical red bars) output by MergeLevels. Note that while predicting all the ground truth aberrations correctly, MergeLevels predicts six additional aberrated regions, including two large loss regions near the ends of each chromosome arm. MergeLevels does not produce probabilistic output so we fix predicted aberrations at probability = 1 and all other locations at probability = 0 for comparative purposes. Panel **C** shows the output of the Base-HMM. The green curve indicates the marginal probability of gain at each location, the red curve indicates 1 minus the marginal probability of loss at each location (left axis). There are numerous false positive predictions with the Base-HMM, many of which are caused by single clone outliers. Panel **D** shows the output of the LSP-HMM (*pooled* mode) with green and red the same quantities as in panel **C** and purple stars indicating the set of predicted outliers. The LSP-HMM predicts all ground truth aberrations correctly and there are much fewer clones falsely predicted as aberrated compared to both MergeLevels and the Base-HMM. Note that the locations of the predicted outliers overlap many of the falsely predicted single clone aberrations by the Base-HMM. Notably, there are several outliers predicted in the leftmost loss region on the p-arm of the chromosome. These correspond to CNPs and therefore alert the user that the significance of this region of loss should be carefully considered.

analysed and published by deLeeuw *et al*. [4]. The data was generated using the Sub-Megabase Resolution Tiling (SMRT) arrays [13] using a set of approximately 32,000 clones that cover the human genome. We normalised the data published in deLeeuw *et al*. [4] according to the stepwise method described in Khojasteh *et al*. [15]. The normalized data was then manually labeled by identifying contiguous regions of gains and losses and then labeling the clones contained in the regions as gains or losses. This 'ground truth' labeling allowed us to test our model on high resolution real data, likely to contain CNPs. Only the autosomes (chromosomes 1

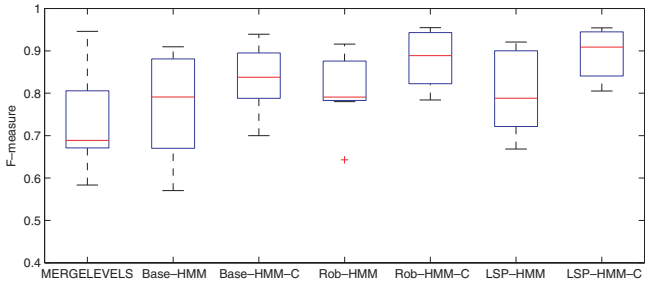


Fig. 4. Distributions of F-measures over eight MCL cell lines for MergeLevels, the Base-HMM, the Rob-HMM and the LSP-HMM with the CNP location prior in *single* and *pooled* mode (labeled with '-C'). All HMM variants performed better than MergeLevels (mean F-measure 0.73 ± 0.11). The LSP-HMM-C variant had the highest mean F-measure (0.89 ± 0.05), followed by the Rob-HMM-C (0.88 ± 0.06), followed by Base-HMM-C (0.84 ± 0.07). In all cases, *pooled* mode outperformed *single* mode.

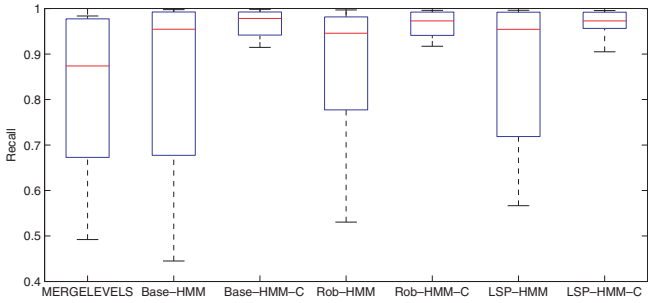


Fig. 5. Distributions of recall over eight MCL cell lines for MergeLevels, the Base-HMM, the Rob-HMM and the LSP-HMM with the CNP location prior in *single* and *pooled* mode (labeled with '-C') . All HMM variants had higher recall rates than MergeLevels (mean recall rate 0.82 ± 0.18). In all cases, *pooled* showed considerable improvement over *single* mode and showed very high recall rates. For Base-HMM-C, Rob-HMM-C and LSP-HMM-C the recall rates were the same at 0.97 ± 0.03.

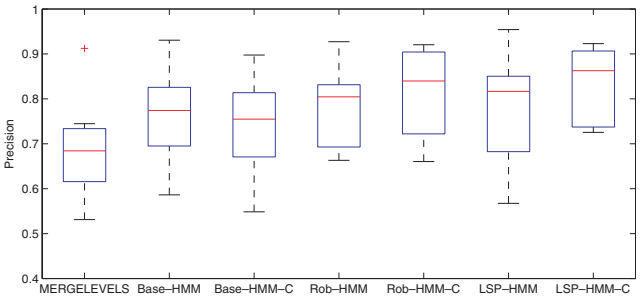


Fig. 6. Distributions of precision over eight MCL cell lines for MergeLevels, the Base-HMM, the Rob-HMM and the LSP-HMM with the CNP location prior in *single* and *pooled* mode (labeled with '-C'). Base-HMM had higher precision (0.76 ± 0.10) which was actually worsened slightly by pooling (Base-HMM-C: 0.74 ± 0.10). The Rob-HMM and Rob-HMM-C had precision of 0.78 ± 0.09 and 0.81 ± 0.09 indicating that robustness and robustness with pooling improves precision over the base model. Finally the LSP-HMM-C had the highest precision rates (0.83 ± 0.08). Pooling for the LSP-HMM showed the most benefit of all the HMM variants.

to 22) contained ground truth labeling therefore only these chromosomes were considered in our analysis. This reduced the number of clones per sample to 29,992. The data set contained a total of 195 aberrated regions: 123 losses and 72 gains covering approximately 1% of the human genome.

We used a list of CNPs (Wong *et al*., unpublished) detected using SMRT arrays on a population of 95 normal individuals to set the LSP probability of an outlier. The list contains all the observed CNPs in the population. We discuss the potential use of other available CNP lists in Section 4.

Figure 3 shows chromosome 1 of HBL-2 with ground truth labels (A), with MergeLevels predictions (B) with Base-HMM predictions (C) and with LSP-HMM predictions (D). The Base-HMM and the LSP-HMM were both run in *pooled* mode. The LSP-HMM was given the complete list of CNPs described above that covered approximately 20% of the clones. We used $p_o = 10\%$ and $t_o = 0.01$ to determine outliers. Other parameters used for this data set are listed in Table 1. For MergeLevels, red bars indicate predicted regions of loss, green bars indicate predicted regions of gain. For the HMMs, red indicates 1 minus the probability of loss and green indicates probability of gain. These plots are similar in spirit to Engler *et al*. [6]. Figure 3 shows that the LSP-HMM predicts all of the aberrated clones with far fewer false positive predictions than both MergeLevels and the Base-HMM. Interestingly, MergeLevels and the Base-HMM are prone to different kinds of false positive predictions. MergeLevels tends to mis-label a small number of large segments with means slightly different than the neutral state mean, whereas the Base-HMM mis-labels a large number of very short segments usually corresponding to outliers. The LSP-HMM is relatively immune to both problems. In addition, the panel (D) depicts predicted outliers as purple stars, showing the qualitative advantage of providing additional information to the user in the output. This is particularly relevant to the left-most loss region in the p-arm. The ground truth labeling actually contains several clones that overlap CNPs. These clones are labeled as outliers by the LSP-HMM and therefore can instruct the user to treat the predicted loss with some degree of caution. In addition to this qualitative assessment of our algorithm, Figure 4 shows distributions of F-measures over the eight MCL cell lines. Distributions are shown as box-and-whisker plots where the line within the box indicates the median of the distribution, the top and bottom edges of the box indicate the third and first quartiles, the ends of the whiskers indicate the 95% confidence intervals of the distribution. The single point shown for Rob-HMM is outside the 95% confidence interval. The distributions show systematic improvement of the Base-HMM over MergeLevels, Rob-HMM over the Base-HMM and the LSP-HMM over the Rob-HMM. MergeLevels had a mean F-measure of 0.73 ± 0.10. Base-HMM had an F-measure of 0.77 ± 0.12 indicating that using an HMM framework improves accuracy over MergeLevels. Further gains were obtained by running the Base-HMM in *pooled* mode (F-measure for Base-HMM-C was 0.84 ± 0.07). Adding robustness in *pooled* mode (Rob-HMM-C) contributed additional improvement (F-measure was 0.88 ± 0.06). Finally using the robust model in *pooled* mode combined with prior knowledge on locations of CNPs (LSP-HMM-C) resulted in the highest accuracy (F-measure was 0.89 ± 0.05). In Figure 4, we can easily see from the boxplots that Base-HMM-C, Rob-HMM, Rob-HMM-C and LSP-HMM-C are all significantly

better (at the 5% level) than MergeLevels. Base-HMM and LSP-HMM are not. We also performed a one way anova test (which is slightly less robust), and found that Rob-HMM-C and LSP-HMM-C are both significantly different (at the 5% level) to MergeLevels. Similar comments apply to the results on simulated data (see Section 3.2). Although the LSP-HMM-C is basically the same as the Rob-HMM-C, it is notable that it does not do worse despite being 'informed' by 20% of the locations in the LSP. This suggests that the model is robust to LSPs that are not supported by the data. This is significant given that our CNP list covers about 20% of the clones, yet in any one sample a much smaller portion of clones are expected to overlap a CNP (recall that the CNP list is made up of a union of all observed CNPs from a population of individuals). We note that the *pooled* mode worked considerably better for all HMM variants, demonstrating the advantage of ''borrowing statistical strength'' from all the data in the sample during parameter estimation.

To assess what was contributing to the differences in F-measure, we plotted precision and recall separately. The recall rates are shown in Figure 5 and demonstrate that pooling shows considerable improvement over *single* mode for the HMM variants. The recall rates were equally very high for the *pooled* HMM variants (0.97 ± 0.03). In contrast, differences in the HMM variants were observed for precision (see Figure 6). We observed improved precision of Rob-HMM-C over the Base-HMM-C indicating that considering outliers reduced the number of false positives. LSP-HMM-C had the highest precision (0.83 ± 0.09), therefore the CNP knowledge further reduced false positives (see Figure 6). The high recall rates for LSP-HMM-C suggests that any future effort to improve accuracy should first focus on reducing false positive predictions to improve precision. However, we noted numerous examples, such as at the centromeric end of the q-arm of HBL-2 chromosome 1 (Figure 3 **D**) where the falsely predicted aberrations could indeed be real.

3.2 Simulated data with outliers

To validate our model on additional data set with ground truth, we used the synthetic data created by Willenbrock and Fridlyand [24], downloaded from http://www.cbs.dtu.dk/~hanni/aCGH/. This data is fairly realistic, since it is generated by sampling segments from a large set of primary tumours [24]. To simulate CNPs, we modified this data by adding outliers planted randomly at 10% of the locations in the samples. The positions were sampled from a uniform distribution from 1 to 2000 (the number of clones in each sample). The $\log_2$ ratios for these outliers were sampled from a Gaussian distribution with mean 0 and variance 2. This gave us a data set with ground truth locations for the aberrated clones and for the positions of the outliers.

We chose 10% as the outlier fraction for the following reason. Our internally generated list of CNPs covers nearly 20% of the SMRT clones. However, publicly available CNPs represent approximately 1% of the SMRT clones. Therefore, we chose 10% as a reasonable compromise between these extremes. We also ran the Base-HMM and Rob-HMM on the original synthetic data and both performed extremely well (mean F-measure 0.95 ± 0.10 and 0.93 ± 0.12 respectively). This provided further justification to create a harder data set that contained the outliers.

In our experiments, we compared the effects of considering all the known outliers to adding additional locations to the prior which were not outliers. This simulated the effect of an incorrect prior. Note that we can choose the strength of the prior. We set the prior probability to 0.01.

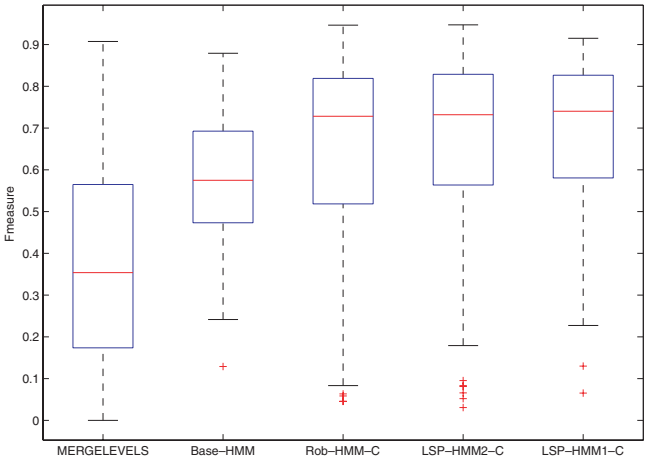


Fig. 7. F-measures for 100 samples of Willenbrock and Fridlyand's simulated data augmented with outliers. From left to right: MergeLevels had an F-measure of 0.37 ± 0.26. The Base-HMM had better accuracy (F-measure of 0.58 ± 0.16). Further improvement was gained with the Rob-HMM-C (F-measure $= 0.64 \pm 0.24$). As expected, informing the LSP-HMM with the locations of the outliers (LSP-HMM1-C) resulted in the best performance. LSP-HMM2-C (F-measure $= 0.66 \pm 0.22$) was informed with a superset of the outlier locations, and LSP-HMM1-C (F-measure 0.68 ± 0.19) was given all and only the outlier locations.

Figure 7 shows the distributions of accuracy on 100 samples for the three variants of our algorithm, including the LSP-HMM informed by a superset of the positions, and exactly all the positions of known outliers. Results on this data echo our results on MCL. MergeLevels performs considerably worse than all the HMMs: its F-measure was 0.37 ± 0.26 over 100 samples. The Base-HMM had a F-measure of 0.58 ± 0.16, validating that by using an HMM framework, significant improvement is attained over MergeLevels. As for MCL data, further improvement was attained by adding outlier detection. The Rob-HMM-C had a F-measure of 0.64 ± 0.24. Finally the versions of the LSP-HMM-C performed better when informed by a superset of the positions (F-measure$=0.66 \pm 0.22$), and exactly all the positions (F-measure 0.68 ± 0.19) of the known outliers. This indicates that a weak prior, when supported by the data can help discover outliers, however contradictory evidence will usually overwhelm the prior when it is wrong.

4 DISCUSSION

We have presented a new model for classifying aberrated clones in aCGH data, which is robust to outliers and is able to leverage prior knowledge about CNP locations. We have demonstrated that on real and simulated data this model works better than a standard HMM and a state of the art method, DNAcopy+MergeLevels. We also determined that estimating parameters of the HMM using pooled data across chromosomes improves accuracy.

Our results showed that recall rates were very high for all HMM variants on the MCL data, and the differences in performance can be mainly attributed to precision rates. We showed that the LSP-HMM

is immune to falsely predicting large regions that MergeLevels typically will mis-label and single clone outliers which the standard HMM falsely predicts as gains. We also showed qualitatively how the LSP-HMM enables the user to cross-reference predicted outliers with known CNPs and therefore allows for a more thorough interpretation of any predicted gains and losses.

As mentioned previously, the hyperparameters for both MCL and the synthetic data were set by hand. We believe that sensitivity to parameter settings (in particular with LSPs) are partially mitigated by pooling data across chromosomes. We showed in Figure 5 and Figure 6 how pooling improved both recall and precision rates for the LSP-HMM. We also noted that even though the CNP list for the MCL data consisted of 20% of the clones, the data overwhelms the prior at most locations. In *pooled* mode this phenomenon is significantly more pronounced as there is substantially more data available to help overwhelm the prior in locations where it is wrong. To further test this theory, our future work will involve accumulating a set of CNPs that is a union of numerous sets of previously published CNPs, for example Iafrate *et al*. [12], Sebat *et al*. [22], Tuzun *et al*. [23], Conrad *et al*. [3] and McCarroll *et al*. [17]. We anticipate that as long the the prior is not too strong a more comprehensive list of LSPs will further help aCGH analysis and the interpretation of results.

In addition to pooling, we plan to add levels to the hierarchy of the model to make it robust to parameter settings. We will put hyper-hyperparameters on the hyperparameters as discussed in Section 2.3. This increases the number of parameters to estimate, but the potential benefits of avoiding hand-tuning of parameters offset this additional cost. In addition, we also set the number of states of the HMM by hand. We noticed that the 4-state model performed better than the 3-state model, however the variance on the 4th state always converged to high values. This allowed the 4th state to 'compete' with the outlier process to explain the outliers, and therefore may have resulted in false positives. We are currently working on a new model that solves the ambiguities observed between high-variance states and the outlier process.

To evaluate the clinical applicability of our model, we plan to apply the method to samples extracted from a cohort of lymphoma patients. The aCGH profiles for these patients have been manually classified and numerous clinically relevant aberrations have been identified. We are also developing new models to identify locations of recurrent aberrations across samples, and to use other forms of prior knowledge, such as the locations of fragile sites. Combined with CNP information, we anticipate that such models will be extremely useful in profiling sub-types of cancer with aCGH.

ACKNOWLEDGEMENTS

This work is supported by a research grant from Genome Canada. SPS is supported by the Michael Smith Foundation for Health Research. We thank Kendi Wong from the British Columbia Cancer Research Centre for help with CNP determination and Kevin Leyton-Brown from the UBC Computer Science department for providing computing resources.

REFERENCES

[1] P Broët and S Richardson. Detection of gene copy number changes in cgh microarrays using a spatially correlated mixture model. *Bioinformatics*, Feb 2006.

[2] P G Buckley, K K Mantripragada, A. Piotrowski, T Diaz de Ståhl, and J P Dumanski. Copy-number polymorphisms: mining the tip of an iceberg. *Trends Genet*, 21(6):315–317, Jun 2005

[3] D F Conrad, T D Andrews, N P Carter, M E Hurles, and J K Pritchard. A high-resolution survey of deletion polymorphism in the human genome. *Nat Genet*, 38(1): 75–81, Jan 2006.

[4] R J de Leeuw, J J Davies, A Rosenwald, G Bebb, R D Gascoyne, M J Dyer, L M Staudt, J A Martinez-Climent and W L Lam. Comprehensive whole genome array cgh profiling of mantle cell lymphoma model genomes. *Hum Mol Genet*, 13(17): 1827–1837, Sep 2004.

[5] P H Eilers and R X de Menezes. Quantile smoothing of array CGH data. *Bioinformatics*, 21(7): 1146–1153, Apr 2005.

[6] D A Engler, G Mohapatra, D N Louis, and R A Betensky. A pseudolikelihood approach for simultaneous analysis of array comparative genomic hybridizations (acgh). *Biostatistics*, Jan 2006.

[7] J Fridlyand, A Snijders, D Pinkel, D Albertson, and A Jain. Hidden Markov Models approach to the analysis of array CGH data. *Journal of Multivariate Statistics*, 90: 132–153, 2004.

[8] A Gelman, J Carlin, H Stern, and D Rubin. *Bayesian data analysis*. Chapman and Hall, 2004. 2nd edition.

[9] S Guha, Y Li, and D Neuberg. Bayesian hidden markov modeling of array cgh data. Technical report, Harvard School of Public Health, 2006.

[10] L Hsu, S G Self, D Grove, T Randolph, K Wang, J J Delrow, L Loo, and P Porter. Denoising array-based comparative genomic hybridization data using wavelets. *Biostatistics*, 6(2): 211–226, Apr 2005.

[11] P Hupé, N Stransky, J Thiery, F Radvanyi, and E Barillot. Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20(18):3413–3422, Dec 2004.

[12] A J Iafrate, L Feuk, M N Rivera, M L Listewnik, P K Donahoe, Y Qi, S W Scherer, and C. Lee. Detection of large-scale variation in the human genome. *Nat Genet* 36(9):949–951, Sep 2004.

[13] A S Ishkanian, C A Malloff, S K Watson, R J DeLeeuw, B Chi, B P Coe, A Snijders, D Albertson, D Pinkel, M Marra, V Ling, C MacAulay, and W Lam. A tiling resolution DNA microarray with complete coverage of the human genome. *Nat Genet*, 36(3):299–303, Mar 2004.

[14] K Jong, E Marchiori, G Meijer, A V Vaart, and B Ylstra. Breakpoint identification and smoothing of array comparative genomic hybridization data. *Bioinformatics* 20(18):3636–3637, Dec 2004.

[15] M Khojasteh, W L Lam, R K Ward, and C MacAulay. A stepwise framework for the normalization of array cgh data. *BMC Bioinformatics*, 6:274–274, Nov 2005.

[16] W R Lai, M D Johnson, R Kucherlapati, and P J Park. Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics* 21(19):3763–3770, Oct 2005.

[17] S A McCarroll *et al*. Common deletion polymorphisms in the human genome. *Nat Genet* 38(1):86–92, Jan 2006.

[18] A B Olshen, E S Venkatraman, R Lucito, and M Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics* 5(4):557–572, Oct 2004.

[19] F Picard, S Robin, M Lavielle, C Vaisse, and J J Daudin. A statistical approach for array CGH data analysis. *BMC Bioinformatics* 6(1):27–27, Feb 2005.

[20] D Pinkel and D G Albertson. Array comparative genomic hybridization and its applications in cancer. *Nat Genet* 37Suppl: 11–17, Jun 2005.

[21] S Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 2002.

[22] J Sebat, B Lakshmi, J Troge, J Alexander, J Young, P Lundin, S Månér, H Massa, M Walker, M Chi, N Navin, R Lucito, J Healy, J Hicks, K Ye, A Reiner, T C Gilliam, B Trask, N Patterson, A Zetterberg, and M Wigler. Large-scale copy number polymorphism in the human genome. *Science*, 305(5683):525–528, Jul 2004.

[23] E Tuzun, A J Sharp, J A Bailey, R Kaul, V A Morrison, L M Pertz, E Haugen, H Hayden, D Albertson, D Pinkel, M V Olson, and E E Eichler. Fine-scale structural variation of the human genome. *Nat Genet*, 37(7):727–732, Jul 2005.

[24] H Willenbrock and J Fridlyand. A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics*, Sep 2005.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e440–e445
doi:10.1093/bioinformatics/btl211

Relative contributions of structural designability and functional diversity in molecular evolution of duplicates

Boris E. Shakhnovich*
Bioinformatics Program, 24 Cummington st., Boston University, Boston MA 02215

ABSTRACT

Analysis of increasingly saturated sequence databases have shown that gene family sizes are highly skewed with many families being small and few containing many, far-diverged homologs. Additionally, recently published results have identified a structural determinant of mutational plasticity: designability that correlates strongly with gene family size. In this paper, we explore the possible links between the two observations, exploring the possible effect of designability on duplication and divergence.

We show that designability has an inverse of expected relationship with strength of selection. More designable domains that should have more mutational plasticity evolve slower. However, we also present evidence that recently duplicated genes have variable probability of locus fixation correlated with strength of selection. As expected, paralogs under stronger evolutionary pressure have a lower failure rate. Finally, we show that probability of pseudogene formation from gene duplication can be directly tied to designability and functional flexibility of the family.

We present evidence that gene families with higher designability have diverged farther because of lower probability of pseudogenization. Additionally, mutational plasticity may play an integral role by influencing pseudogenization rate. Either way, we show that considering the failure rate of duplications is integral in understanding the determinants and dynamics of molecular evolution.

Contact: borya@acs.bu.edu

1 INTRODUCTION

Evolution progresses through an iterative process of duplication and differentiation. The fate of genes that have undergone duplication was hypothesized by Ohno (Ohno, 1970) to fall into one of three categories: (i) Neo-functionalization when the newly duplicated gene evolves under purifying selection to acquire novel function, (ii) Non-functionalization when the newly duplicated gene becomes non-functional through accumulation of deleterious mutations and (iii) sub-functionalization when both copies mutate to divide ancestral pleiotropy. (Kondrashov, *et al*., 2002; Petrov and Hartl, 2000) In order to understand the driving forces behind duplication and divergence Kimura and Ota in their seminal paper from 1974 (Kimura and Ota, 1974) outlined the principles governing the progress of molecular evolution. One of the principles described in that paper is ''evolution by mutational pressure''. This principle outlines the strong preferential elimination of deleterious mutants e.g. those that destroy the structure or function. Kimura and Ota go so far as to suggest that it is the pressure on the deleterious mutants more than positive pressure canonical to Darwinian evolution that is the predominant force driving evolution of multigene families (Kimura and Ota, 1974).

Driven by the recent availability of sequenced genomes coupled with high-throughput functional assays, researchers observed a number of significant correlations between intrinsically functional characteristics of gene sequences such as essentiality (Hirsh and Fraser, 2001; Hurst and Smith, 1999; Yang, *et al*., 2003), number of protein-interaction partners (Fraser, *et al*., 2002), or expression level (Drummond, *et al*., 2005; Pal, *et al*., 2001), and the strength of purifying selection. However, the relative importance of each characteristic has been a subject of vigorous debate. (Drummond, *et al*., 2005; Drummond, *et al*., 2006; Jordan, *et al*., 2003; Wall, *et al*., 2005) While the process of duplication and differentiation is well documented (Haldane, 1933; Ohno, 1970), gene-specific characteristics that influence this process are largely unknown.

Finally, there is considerable debate concerning the relationships between the strength of selection and subsequent dynamics of gene duplication and divergence. Recently, several researchers found that duplication occurs under influence of purifying selection (Shiu, *et al*., 2006). Newly duplicated genes undergo a brief period of relaxation followed immediately by increase of selection. (Conant and Wagner, 2003; Jordan, *et al*., 2004; Kondrashov, *et al*., 2002; Zhang, *et al*., 2003). On the other hand, survivorship curves for worm duplicates show that simple stochastic models considering only the age of the duplicate pair do equally well at predicting distributions of duplicates (Lynch and Conery, 2000). The unifying characteristic of these models lies in their attempt to understand the determinants of duplication success and forego the evaluation of the equally important duplication failure rate. Since the most frequent outcome of duplication is non-functionalization of at least one paralog (Nei, 1973; Petrov and Hartl, 2000), understanding that process is fundamental to relating gene function, selection and duplication.

While we showed that protein structure correlates with the divergence of homologs (Shakhnovich, *et al*., 2005), there have been no investigations into the relationships between protein structure,

selection and the dynamics of duplication and divergence. Thus, in this work we evaluate the relationships between designability, selection and rate of pseudogenization. First, we show that gene pairs under stronger purifying selection have a lower probability of pseudogenization. We then show that that structural determinants of mutability e.g. designability (England, *et al*., 2003; England and Shakhnovich, 2003) inversely correlates with strength of purifying selection. Since we previously established the link between divergence and designability, we explain this link here by showing that higher designabiltity domains enjoy a lower pseudogenization rate. Consistent with previous results, we find that the lower pseudogenization rate of higher designability domains results in farther diverged paralogs. These observations carry fundamental implications for understanding and application of Kimura's theory (Kimura and Ota, 1974) of evolution by mutational pressure.

2 STRUCTURAL DETERMINANTS OF SELECTION

The strength of selection on the paralogs after duplication may be influenced by many factors such as lethality (Venkatesan, *et al*., 2003; Yang, *et al*., 2003), paralogy (Jordan, *et al*., 2004), ability to neo-functionalize (Lynch and Conery, 2000; Lynch, *et al*., 2001) and protein interaction neighbors (Fraser, *et al*., 2003). We were interested in probing whether structural properties of domains have significant impact on evolutionary pressure. Recent theoretical investigations showed that designability can be used a measure of mutational plasticity allowable for a structure. While measures for designability were first derived using lattice models from theoretical arguments of stability (England and Shakhnovich, 2003; Shakhnovich, *et al*., 2005; Shakhnovich and Max Harvey, 2004), recent investigations have seen a generalization that enabled calculation of designability on real protein structures. Specifically, England and Shakhnovich showed that contact density (CD) of structure (England and Shakhnovich, 2003) can serve as an approximation to a determinant of protein designability: domains with higher CD tend to be more designable.

Indeed, when investigating real protein domains, we found that those with higher CD tend to have farther diverged homologs (Shakhnovich, *et al*., 2005). We also found that designability is a potential for sequence diversity. This potential has to be fulfilled through an iterative process of duplication and divergence in each genome. Since designability is related to mutational plasticity, it may also influence the evolutionary pressure on the recently duplicated paralogous pair. Thus, the null hypothesis would be that designability relaxed evolutionary pressure.

To test the influence of designability on evolutionary pressure, we compare CD with the strength of purifying selection. Evolutionary pressure can be measured by computing the ratio of accepted non-synonymous substitutions K_a to the synonymous substitution rate K_s between homologous sequences (Gaut and Doebley, 1997; Hughes and Hughes, 1993; Li, 1997). We used the widely used PAML package to estimate the Ka and Ks ratios (Yang and Nielsen, 2000). Duplicates under stronger evolutionary pressure exhibit K_a/K_s ratios close to 1. Deviations from 1 are usually taken to mean that the paralogs are under relaxed pressure.

We found that Ka/Ks ratio of recently duplicated gene pairs in *C. Elegans* (Lynch and Conery, 2000) correlate inversely with the theoretical determinant of designability: the CD of the structure. To further test the robustness of this relationship we performed the same calculation on orthologs from S. Cerevisiae and Paradoxus. We found the same general trend of higher designability domains under stronger evolutionary pressure (Fig 1). This is a surprising result since higher designability should impart a more relaxed mutational regime. This result is also surprising in light of previous evidence that more diverse gene families encode more designable domains. (Shakhnovich, *et al*., 2005) Thus, it seems that paralogs encoding more designable domains are under stronger selection, but diverge farther away to yield gene families with higher sequence entropy. We attempt to reconcile these by looking at the effect of stronger selection on duplication and divergence.

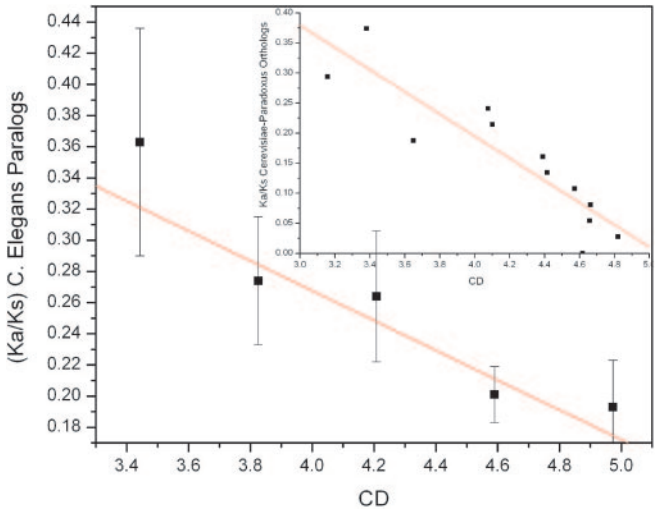


Fig. 1. Correlation of designability with strength of selection for *C. Elegans* paralogs. CD was calculated (see Methods) as the average over all domains encoded by a gene with more than one domain. Each bin contains between 20-25 genes. The red line indicates a linear fit to the scatter plot R=.91. The error bars represent variance inside each bin. The same correlation was calculated for S. Cerevisiae-Paradoxus orthologs with similar results (inset). Red line is the linear fit R=.92, P<.001.

3 HOW SELECTION AFFECTS PSEUDOGENIZATION RATE

Thus, it seems that designable domains evolve slower, but have farther diverged families. One possible explanation is that domains under relaxed evolutionary pressure pseudogenize more often. Thus, these families may not have a chance to diverge as functional paralogs leading to more compact families with less sequence entropy. We can conjecture, based on evidence from previous work (Conant and Wagner, 2003; Kondrashov, *et al*., 2002; Zhang, *et al*., 2003) that mutation and diversification of a newly duplicated locus may be related to the strength of selection on the paralogous pair. In order to assess the influence of evolutionary pressure on duplication events, we consider the relationship between selection and failure of a duplication event e.g. pseudogene formation (Harrison and Gerstein, 2002; Harrison, *et al*., 2002; Kimura and Ota, 1974). We test this relationship directly using pseudogenes identified in a number of recently sequenced genomes.

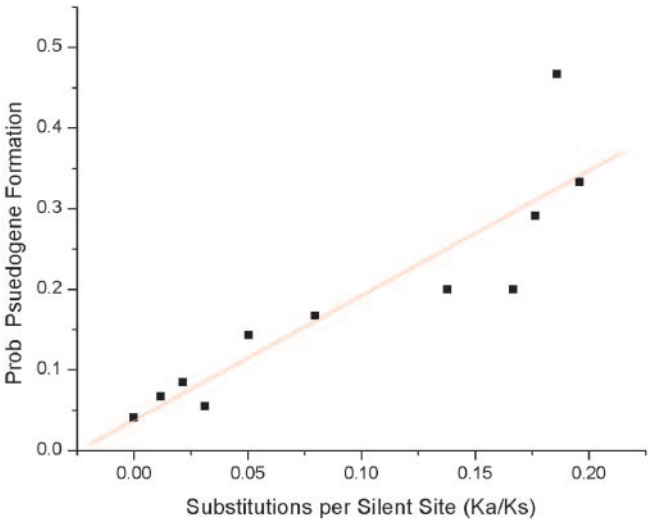


Fig. 2. The average probability of pseudogenization (P_{pseudo}) versus strength of selection as estimated by K_a/K_s. The data was binned on the X axis with step .01. In case no pseudogenes were found the multigene families were not considered for this plot. The red line is the linear fit with R = .89 and P<1e-4. The relatively small number of multigene families with identifiable pseudogenes prevents reliable analysis of statistical robustness of the fit.

Recently Gerstein and coworkers have identified all putative pseudogenes in a number of genomes (Harrison, *et al*., 2002; Harrison, *et al*., 2001; Harrison and Gerstein, 2002; Harrison, *et al*., 2002). Pseudogenes are disabled copies of genes (or decayed remnants of genes) that do not produce a full-length protein chain. Pseudogenes can generally be divided into two types. Processed pseudogenes arise from reverse transcription from messenger RNA (mRNA) and re-integration into the genomic DNA. By contrast, ''duplicated pseudogenes'' arise from duplication in the genomic DNA and subsequent disablement, most commonly through disruptive frameshift mutation or premature stop codon formation. Clearly we want to focus on this class of pseudogenes as a measure of locus fixation failure.

We start our analysis of the differential failure rate of duplication by calculating the probability of successful duplication versus pseudogenization of one of the duplicates in *C. Elegans*. We employ simple sequence comparison methods to identify paralogous members of multigene families and pseudogenes likely formed from a duplication event of any member in those families. (See Methods) In this paper, the probability of pseudogene formation is just the ratio of the number of pseudogenes weakly homologous to a multigene family versus the number of paralogs belonging to the same multigene family. (See Methods) This ratio may not be direct evidence of variable probability of locus fixation, since failure may not always lead to pseudogene formation. It may, however, be used as a first-pass approximation given the assumption that deletion of pseudogenes from the genome is nearly uniform, i.e. without preference to individual duplication locii.

We first plot average K_a/K_s(Lynch and Conery, 2000) of *C. Elegans* paralogs against P_{pseudo} (Harrison, *et al*., 2001). Figure 2 shows that recently duplicated paralogous pairs under strong selection are less likely to leave pseudogenes. Thus, higher K_a/K_s ratios have higher P_{pseudo}. This is an intuitively obvious result underlining the relationship between evolutionary pressure and duplication success. Of course, the absence of pseudogenes does not imply success. However, controlling for age of duplicates does not qualitatively change results. Additionally, we observe no difference in the duplication rate of paralogs. (data not shown) Unfortunately, the number of pseudogenes is relatively small and thus we can calculate only average behavior and the its impossible to relate designability and duplication directly using data from *C. Elegans*.

The insight here is that duplication and differentiation is a stochastic process. This process has a certain success and failure rate. Accumulation of deleterious mutations will increase the chance that the protein becomes non-functional and thus turn into a pseudogene. This is consistent with the postulate of evolution by mutational pressure. Thus, the implication from Fig 1 and 2 is that more designable domains, under stronger selection are able to avoid accumulation of deleterious mutations and thus fewer duplicates pseudogenize. This could, potentially, result in farther diverged families, despite the relatively slower rate of divergence.

4 STRUCTURAL AND FUNCTIONAL DETERMINANTS OF DUPLICATION FAILURE

We saw in the previous two sections how structure can affect the dynamics of duplication and diversification through increased selection on higher designability domains. Along with considerations of structural stability, we can hypothesize that dynamics of molecular evolution may be affected by the functional landscape of gene family. Some recently duplicated paralogs may have finding new functions with fewer mutations. (Shakhnovich, *et al*., 2003) If neo-functionalization is easier, the duplicate is more likely to be fixed in the population through positive selection. In fact, we previously reported the correlation between designability and functional diversity (Shakhnovich, *et al*., 2005). Thus, to test the relationship between designability and dynamics of molecular evolution, we have to take into account both the positive selection due to neo-functionalization and structural determinants of mutational plasticity.

We begin by calculating the probability landscape of pseudogene formation with respect to designability and the potential for functional diversity (FFS) in the *M. Leprae* (Smith, *et al*., 1997) genome. We calculate the contact density (CD) of each gene using homologous structures in PDB. The potential for functional flexibility (FFS) is calculated using GO annotations (Shakhnovich, *et al*., 2005). All homologous sequences in a non-redundant database (NRDB) are annotated on GO (Ashburner, *et al*., 2000), the entropy of annotation on each level is the FFS (See Methods).We chose the *M. Leprae* genome mainly for statistical reasons, because it has the largest number of pseudogenes of any prokaryotes (Cole, *et al*., 2001; Smith, *et al*., 1997).

The pseudogenization probability landscape shown in Fig. 3 shows that the fixation of the duplication event in *M. Leprae* is a function of both the structural designability (CD of the structure) and functional diversity of the protein domain. Duplicates of genes coding for the most highly designable domains with the higherst potential for functional diversity have the least chance of becoming pseudogenes. The fact that the maximum probability of pseudogenization is away from the minimum designability is probably due to finite size effects and inherent error in the calculations. The land-

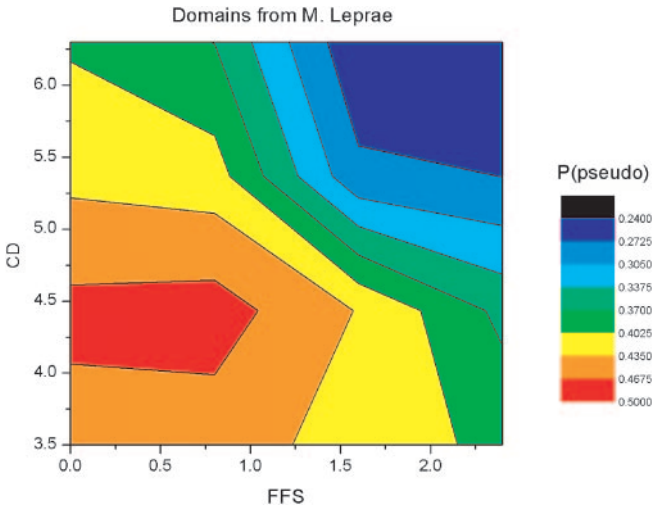


Fig. 3. The probability of pseudogene formation with respect to both the functional flexibility score (FFS) and designability (CD) of the domain. For each domain we calculated the number of paralogs in M. Leprae and the number of duplicated pseudogenes likely to have come from that domain found in the same genome. We then calculated the probability of pseudogene formation and plotted it with respect to CD and FFS of that domain. The plot was made by binning FFS and CD into bins with step .35 and averaging the probability inside each bin. Each bin contains between 20 and 30 domains. It is striking how the probability of successful duplication depends on both the designability of the structure and the functional diversity of the domain. At highest CD and FFS the probability is around 25 percent of failure while at the lowest CD and FFS the probability increases to more than 50 percent. This would suggest that for a given duplication attempt, some domains enjoy more than fifty percent increase in their ability to ''produce offspring''.

scape on Fig 3 shows that both structural and functional characteristics of genes, and more importantly the families to which these genes belong affect the potential success of duplication and divergence of paralogs. These finding are consistent with farther divergence of homologs of more designable domains. (Shakhnovich, *et al*., 2005) While this evidence is suggestive of a link between designability and the dynamics of molecular evolution, it is not necessarily sufficient due to possible transitive relationships not controlled for in this particular study (Drummond, *et al*., 2006).

Some researchers have argued that the *M. Leprae* genome has undergone reductive evolution (Cole, *et al*., 2001; Harrison and Gerstein, 2002; Smith, *et al*., 1997) instead of an explosive period of duplication. However, the principles of Kumura and Ohta e.g. evolution by mutational pressure remain in effect for gene deletion as well as gene accumulation. We can justifiably interpret results in Fig. 3 as the probability of deletion of a member of a multigene family is inversely proportional to the designability of the structure encoded by its members and diversity functions inherent in the members.

5 DISCUSSION

The question of the relationship between functional or genetic characteristics and evolutionary pressure is at the forefront of studies in molecular evolution (Drummond, *et al*., 2005; Drummond, *et al*., 2005; Jordan, *et al*., 2002; Jordan, *et al*., 2003; Jordan, *et al*., 2004). A comprehensive theory of this relationship may lead to a better understanding of the phenotype-genotype relationship. Since we previously reported that mutational flexibility as measured by structural designability correlates with sequence entropy of the gene family (Shakhnovich, *et al*., 2005). However, exploration of sequence space occurs through iterative duplication and differentiation of paralogs in genomes under evolutionary pressure. Thus, we wanted to explore the genetic mechanisms that could result in association of larger sequence families with higher designability structures. We initially hypothesized that genes coding for higher designability domains evolve faster and are thus able to diverge farther.

Contrary to our initial expectations, we found that genes coding for higher designability domains are under stronger selective pressure. However, the fact that they diverge slower doesn't preclude them from diverging farther. (Shakhnovich and Koonin, 2006) To explain the seeming inconsistency, we postulated that there exists an increased success rate of locus fixation for paralogs under stronger evolutionary pressure. Indeed, using data from C. Elegans, we found that paralogs under stronger selection pseudogenize less. We are in the process of exploring this dependency between pseudogenization and selection in more detail in our upcoming publication. (Shakhnovich and Koonin, 2006)

Since we show above that lower evolutionary pressure confers a higher probability of failure upon duplication, we are able to show directly in Fig. 3 that genes encoding domains with higher designability have a lower probability of pseudogenization. In fact, we show that probability of success in locus fixation correlates with both the designability of the structure and the functional flexibility of the gene family. The interpretation is that larger functional flexibility confers a smaller mutational path for neo-functionalization e.g. the paralogs have to accept fewer mutations before finding new function. At the same time, mutational plasticity granted by higher designability structures allows the paralogs more ''time'' before deleterious mutations destroy protein stability. Both of these factors can be seen using the pseudogenization landscape in Figure 3.

The above evidence is a striking observation of the effect of fitness and the mechanism of natural selection on protein domains. Akin to organismal evolution, protein domains undergo cycles of trial and tribulation in the form of stochastic duplication and divergence. Some domains survive the duplication cycle and others turn into pseudogenes. The difference between the domains that survive and those that perish lies in the origin of the duplication event with respect to sequence space. The newly duplicated gene pairs that encode more designable structures have a higher probability of success due to the fact that both copies may accept more mutations and are less likely encounter a deleterious mutation that destroys structure.

One caveat from this study is the assumption of equal duplication rate and pseudogene deletion across genes coding for variable designability domains. Furthermore, there could be other transitive correlations with designability and functional flexibility, not accounted for in this study. For example, more work has to be done in order to determine the underlying reasons for higher selection on higher designability domains. These reasons could be historical e.g. higher designability domains are older (England, *et al*., 2003) or functional e.g. that designable domains code for ''more important'' functions. Moreover, the relative contributions of mutational plasticity and stronger

selection of designable domains in achieving more divergent families is unclear.

A comprehensive model describing the interplay between these parameters is needed for a complete understanding of the effect of structure on dynamics of molecular evolution. While the above are only first studies of the average effect of natural selection on protein domains, it represents a striking confirmation of a paradigm first laid out by Kimura and Ota of evolution by mutational pressure. In that paper Kimura and Ota stated that ability to sustain mutation is the fitness characteristic driving molecular evolution of multigene families. The dependency of pseudogene formation on designability through higher probability of survival of both duplicates opens the door to including structural characteristics in modeling ''fitness'' of the multigene families.

METHODS

The data for K_a/K_s values for pairs of C. Elegans genes was taken directly from (Lynch and Conery, 2000) http://www.csi.uoregon.edu/projects/genetics/duplications/C.elegans.txt. Since this dataset was prepared before the current wormpep database, we excluded genes that were found to be part of pseudogenes, not in the current release of wormpep or were subsequently to Lynch and Conery analysis found to be the same gene. This yielded 1518 duplicate pairs instead of 1770 included in the original file.

To calculate the number of paralogs for each *C. Elegans* gene (N_{dupl}) we performed an all-against all BLAST of the *C-elegans* genome. We took reverse best hits at 1e-6. For each gene we calculated the number of best hits. We also performed BLAST homology search for the genome against all known pseudogenes. The results were used to calculate N_{pseudo}. P_{pseudo} was then calculated using eq2. We also performed a domain-centric calculation of the same quantity. We took all HSSP (Holm and Sander, 1997) domains and counted the number of different ORFs to which they are homologous, this was used for N_{dupl}. We also calculated the number of pseudogenes homologous to the same HSSP(Holm and Sander, 1997) domain and used that for N_{pseudo}. The results were not significantly affected between the two approaches.

To calculate designability for each gene we took all structurally resolved domains from HSSP homologous to the gene with E<1e-6 and more than 40% sequence identity. We then averaged the contact density, if the gene contained more than one domain. Designability of each domain was calculated through contact density as explained in detail below.

As before(Shakhnovich and Max Harvey, 2004), for FFS we calculate the average amount of information per GO(Ashburner, *et al*., 2000) annotation level needed to fully describe the function of each set of sequences that fold into a domain by using the following equation

$$FFS = -\frac{1}{Max(L)} \sum_{l} \sum_{i \in \{\text{nodes on Level } l\}} p_i Log(p_i) \qquad (1)$$

Here, Max(L) is the maximal number of levels of annotation, the summation is taken over all levels l and over all nodes i filled by the functions on the GO (Ashburner, *et al*., 2000) tree, and p_i is the percentage of the sequences that are annotated with function i.

We used data from Gerstein and co-workers who identified all putative duplicated pseudogenes in a number of genomes. (Harrison, *et al*., 2002; Harrison, *et al*., 2001; Harrison and Gerstein, 2002) We downloaded the *M. Leprae* genome from (from Sanger Centre Pathogen Sequencing Group (ftp://ftp.sanger.ac.uk/pub/pathogens/leprae) and *C. elegans* from NCBI because these organisms have the largest number of observed pseudogenes. In contrast to finding functional domains in genomes, the acceptable E value for identifying pseudogenes was lowered to $1e^{-3}$ to signify the increased propensity of pseudogenes to diverge in sequence since pseudogenes are not under pressure to be structurally of functionally viable.

To avoid confusion of which domain corresponds to the failed duplication attempt, we require that there is at most one structure that corresponds to any pseudogene sequence. Thus for each domain we can estimate the probability of becoming a pseudogene using the following simple equation

$$P_{pseudo}^{genefam} = \frac{N_{\text{pseud}}}{N_{\text{pseudo}} + N_{\text{dupl}}} \qquad (2)$$

where N_{pseud} is the number of pseudogenes with high sequence similarity to a domain and N_{dupl} is the number of paralogous genes for the same domain in a given genome.

Designability

England and Shakhnovich showed recently (England and Shakhnovich, 2003) that for a large class of amino acid interaction potentials *B*, the free energy per monomer *f in sequence space* for a protein structure defined by its contact matrix (CM) *C* is given by

$$f = -\frac{1}{N} \sum_{n=2}^{\infty} (\text{Tr } C^n) a_n \qquad (3)$$

where the weights a_i are all positive functions which depend on the interaction energies *B*. The contact matrix C is defined as $C_{ij}=1$ if amino acids i and j are in contact and 0 otherwise. Definitions of contact may vary, but in this paper we use the standard cutoff of 7.5 angstroms between C_β atoms (C_α for Gly). Elementary matrix algebra suggests that trace of high powers of a matrix is determined by its maximal eigenvalue. Thus, protein structures that have greater maximal eigenvalues of their contact matrices are expected to be more designable.

ACKNOWLEDGEMENTS

The author would like to thank Eugene Koonin and Eugene Shakhnovich for their support and insight into the work presented in this manuscript. The work was supported by NIH.

REFERENCES

Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T., Harris,M.A., Hill,D.P., Issel-Tarver,L., Kasarskis,A., Lewis,S., Matese,J.C., Richardson,J.E., Ringwald,M., Rubin,G.M. and Sherlock,G. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, **25**, 25–29.

Cole,S.T., Eiglmeier,K., Parkhill,J., James,K.D., Thomson,N.R., Wheeler,P.R., Honore,N., Garnier,T., Churcher,C., Harris,D., Mungall,K., Basham,D., Brown,D., Chillingworth,T., Connor,R., Davies,R.M., Devlin,K., Duthoy,S., Feltwell,T., Fraser,A., Hamlin,N., Holroyd,S., Hornsby,T., Jagels,K., Lacroix,C., Maclean,J., Moule,S., Murphy,L., Oliver,K., Quail,M.A., Rajandream,M.A., Rutherford,K.M., Rutter,S., Seeger,K., Simon,S., Simmonds,M., Skelton,J., Squares,R., Squares,S., Stevens,K., Taylor,K., Whitehead,S., Woodward,J.R. and Barrell,B.G. (2001) Massive gene decay in the leprosy bacillus. *Nature*, **409**, 1007–1011.

Conant,G.C. and Wagner,A. (2003) Asymmetric sequence divergence of duplicate genes. *Genome Res*, **13**, 2052–2058.

Drummond,A.D., Raval,A. and Wilke,C.O. (2005) A single determinant for the rate of yeast protein evolution, *arxiv.org*. .

Drummond,D.A., Bloom,J.D., Adami,C., Wilke,C.O. and Arnold,F.H. (2005) Why highly expressed proteins evolve slowly. *Proc Natl Acad Sci USA*, **102**, 14338–14343.

Drummond,D.A., Raval,A. and Wilke,C.O. (2006) A single determinant dominates the rate of yeast protein evolution. *Mol Biol Evol*, **23**, 327–337.

England,J.L., Shakhnovich,B.E. and Shakhnovich,E.I. (2003) Natural selection of more designable folds: a mechanism for thermophilic adaptation. *Proc Natl Acad Sci USA*, **100**, 8727–8731.

England,J.L. and Shakhnovich,E.I. (2003) Structural determinant of protein designability. *Phys Rev Lett*, **90**, 218101.

Fraser,H.B., Hirsh,A.E., Steinmetz,L.M., Scharfe,C. and Feldman,M.W. (2002) Evolutionary rate in the protein interaction network. *Science*, **296**, 750–752.

Fraser,H.B., Wall,D.P. and Hirsh,A.E. (2003) A simple dependence between protein evolution rate and the number of protein-protein interactions. *BMC Evol Biol*, **3**, 11.

Gaut,B.S. and Doebley,J.F. (1997) DNA sequence evidence for the segmental allotetraploid origin of maize. *Proc Natl Acad Sci USA*, **94**, 6809–6814.

Haldane,J.B.S. (1933) The part played by recurrent mutation in evolution. *Am Nat*, **67**, 5–19.

Harrison,P., Kumar,A., Lan,N., Echols,N., Snyder,M. and Gerstein,M. (2002) A small reservoir of disabled ORFs in the yeast genome and its implications for the dynamics of proteome evolution. *J Mol Biol*, **316**, 409–419.

Harrison,P.M., Echols,N. and Gerstein,M.B. (2001) Digging for dead genes: an analysis of the characteristics of the pseudogene population in the Caenorhabditis elegans genome. *Nucleic Acids Res*, **29**, 818–830.

Harrison,P.M. and Gerstein,M. (2002) Studying genomes through the aeons: protein families, pseudogenes and proteome evolution. *J Mol Biol*, **318**, 1155–1174.

Harrison,P.M., Hegyi,H., Balasubramanian,S., Luscombe,N.M., Bertone,P., Echols,N., Johnson,T. and Gerstein,M. (2002) Molecular fossils in the human genome: identification and analysis of the pseudogenes in chromosomes 21 and 22. *Genome Res*, **12**, 272–280.

Hirsh,A.E. and Fraser,H.B. (2001) Protein dispensability and rate of evolution. *Nature*, **411**, 1046–1049.

Holm,L. and Sander,C. (1997) Dali/FSSP classification of three-dimensional protein folds. *Nucleic Acids Res*, **25**, 231–234.

Hughes,M.K. and Hughes,A.L. (1993) Evolution of duplicate genes in a tetraploid animal, Xenopus laevis. *Mol Biol Evol*, **10**, 1360–1369.

Hurst,L.D. and Smith,N.G. (1999) Do essential genes evolve slowly? *Curr. Biol*, **9**, 747–750.

Jordan,I.K., Rogozin,I.B., Wolf,Y.I. and Koonin,E.V. (2002) Essential genes are more evolutionarily conserved than are nonessential genes in bacteria. *Genome Res*, **12**, 962–968.

Jordan,I.K., Wolf,Y.I. and Koonin,E.V. (2003) No simple dependence between protein evolution rate and the number of protein-protein interactions: only the most prolific interactors tend to evolve slowly. *BMC Evol Biol*, **3**, 1.

Jordan,I.K., Wolf,Y.I. and Koonin,E.V. (2004) Duplicated genes evolve slower than singletons despite the initial rate increase. *BMC Evol Biol*, **4**, 22.

Kimura,M. and Ota,T. (1974) On some principles governing molecular evolution. *Proc Natl Acad Sci USA*, **71**, 2848–2852.

Kondrashov,F.A., Rogozin,I.B., Wolf,Y.I. and Koonin,E.V. (2002) Selection in the evolution of gene duplications. *Genome Biol*, **3**, RESEARCH0008.

Li,W.H. (1997) *Molecular Evolution*. Sinauer, Sunderland, MA.

Lynch,M. and Conery,J.S. (2000) The evolutionary fate and consequences of duplicate genes. *Science*, **290**, 1151–1155.

Lynch,M., O'Hely,M., Walsh,B. and Force,A. (2001) The probability of preservation of a newly arisen gene duplicate. *Genetics*, **159**, 1789–1804.

Nei,M. and Roychoudhury,A.K. (1973) Probability of fixation of nonfunctional genes at duplicate loci. *Am Nat*, **107**, 590–605.

Ohno,S. (1970) *Evolution by gene duplication*. Springer-Verlag, Berlin, New York.

Pal,C., Papp,B. and Hurst,L.D. (2001) Highly expressed genes in yeast evolve slowly. *Genetics*, **158**, 927–931.

Petrov,D.A. and Hartl,D.L. (2000) Pseudogene evolution and natural selection for a compact genome. *J Hered*, **91**, 221–227.

Shakhnovich,B. and Koonin,E.V. (2006) The Origins and Impact of Constraint in Molecular Evolution of Gene Families. *Genome Res*, In Review.

Shakhnovich,B.E., Deeds,E., Delisi,C. and Shakhnovich,E. (2005) Protein structure and evolutionary history determine sequence space topology. *Genome Res*, **15**, 385–392.

Shakhnovich,B.E., Dokholyan,N.V., DeLisi,C. and Shakhnovich,E.I. (2003) Functional fingerprints of folds: evidence for correlated structure-function evolution. *J Mol Biol*, **326**, 1–9.

Shakhnovich,B.E. and Max Harvey,J. (2004) Quantifying structure-function uncertainty: a graph theoretical exploration into the origins and limitations of protein annotation. *J Mol Biol*, **337**, 933–949.

Shiu,S.H., Byrnes,J.K., Pan,R., Zhang,P. and Li,W.H. (2006) Role of positive selection in the retention of duplicate genes in mammalian genomes. *Proc Natl Acad Sci. USA*, **103**, 2232–2236.

Smith,D.R., Richterich,P., Rubenfield,M., Rice,P.W., Butler,C., Lee,H.M., Kirst,S., Gundersen,K., Abendschan,K., Xu,Q., Chung,M., Deloughery,C., Aldredge,T., Maher,J., Lundstrom,R., Tulig,C., Falls,K., Imrich,J., Torrey,D., Engelstein,M., Breton,G., Madan,D., Nietupski,R., Seitz,B., Mao,J.I. *et al.* (1997) Multiplex sequencing of 1.5 Mb of the Mycobacterium leprae genome. *Genome Res*, **7**, 802–819.

Venkatesan,K., McManus,H.R., Mello,C.C., Smith,T.F. and Hansen,U. (2003) Functional conservation between members of an ancient duplicated transcription factor family, LSF/Grainyhead. *Nucleic Acids Res*, **31**, 4304–4316.

Wall,D.P., Hirsh,A.E., Fraser,H.B., Kumm,J., Giaever,G., Eisen,M.B. and Feldman,M.W. (2005) Functional genomic analysis of the rates of protein evolution. *Proc Natl Acad Sci USA*, **102**, 5483–5488.

Yang,J., Gu,Z. and Li,W.H. (2003) Rate of protein evolution versus fitness effect of gene deletion. *Mol Biol Evol*, **20**, 772–774.

Yang,Z. and Nielsen,R. (2000) Estimating synonymous and nonsynonymous substitution rates under realistic evolutionary models. *Mol Biol Evol*, **17**, 32–43.

Zhang,P., Gu,Z. and Li,W.H. (2003) Different evolutionary patterns between young duplicate genes in the human genome. *Genome Biol*, **4**, R56.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e446–e453
doi:10.1093/bioinformatics/btl235

Integrating image data into biomedical text categorization

Hagit Shatkay*, Nawei Chen and Dorothea Blostein
School of Computing, Queen's University, Kingston, Ontario, Canada

ABSTRACT

Categorization of biomedical articles is a central task for supporting various curation efforts. It can also form the basis for effective biomedical text mining. Automatic text classification in the biomedical domain is thus an active research area. Contests organized by the KDD Cup (2002) and the TREC Genomics track (since 2003) defined several annotation tasks that involved document classification, and provided training and test data sets. So far, these efforts focused on analyzing only the text content of documents. However, as was noted in the KDD'02 text mining contest—where *figure-captions* proved to be an invaluable feature for identifying documents of interest—images often provide curators with critical information. We examine the possibility of using information derived *directly from image data,* and of integrating it with text-based classification, for biomedical document categorization. We present a method for obtaining features from images and for using them—both alone and in combination with text—to perform the triage task introduced in the TREC Genomics track 2004. The task was to determine which documents are relevant to a given annotation task performed by the Mouse Genome Database curators. We show preliminary results, demonstrating that the method has a strong potential to enhance and complement traditional text-based categorization methods.

Contact: shatkay@cs.queensu.ca

1 INTRODUCTION

Categorization of biomedical text is pivotal both for supporting curation tasks in biological databases and for providing researchers with literature appropriate for their specific information needs. For example, curators for the Mouse Genome Database (MGD) need publications with specific contents to validate the expression of genes under certain conditions. Other examples for curation-related task include the identification of papers discussing subcellular localization in support of the annotation of proteins with Gene Ontology (GO) codes for subcellular component, or of papers discussing function—to be used as evidence for functional annotation. On the other side of the quest for information, scientists in individual labs may want to easily identify papers that are likely to be related to their own research, or may look for papers discussing a new area of interest into which they are ready to venture. Underlying all these examples is the need to identify a subset of documents, with some common topical characteristic, within a large set of documents. The latter set may include hundreds of documents returned by a broad PubMed search, or possibly thousands of documents in a certain journal, or even the millions of documents comprising the whole of MEDLINE.

In the past few years several initiatives were established to encourage and evaluate work on biomedical text categorization. The KDD'02 cup (Yeh *et al*., 2003) had a task in which documents were to be categorized as containing (or not containing) evidence for gene expression within the Drosophila wild type, in support of FlyBase curation. For the past two years the TREC Genomics track (Hersh *et al*., 2005, 2006) featured a text categorization task, in which documents were to be classified according to their evidence contents in support of assigning GO annotation to mouse genes. Part of *Task 2* of the BioCreative challenge (Hirschman *et al*., 2005) involved identifying papers that contain evidence for assigning GO codes to human proteins, in support of Swiss-Prot curation.

In all these tasks the documents were categorized based only on the text occurring in them. While participating in the KDD cup, Regev *et al*. (2002) noted that the use of figure captions proved particularly helpful for their high performance in identifying documents discussing gene expression. Following this work, figure captions were also used by participants in the TREC Genomics track (Darwish and Madkour, 2005) as part of the text-features used for categorization. The success of using figure captions is related to the fact that figures contain important cues that are typically used by database curators and annotators to quickly scan documents and distinguish relevant from irrelevant ones. FlyBase curators have indeed indicated that the experimental results shown in papers and used in support of curation, are often presented in figures and their captions (Yeh *et al*., 2003). Figures are often content rich and concisely summarize the most important results or methods used and described in an article.

Our present work is motivated by this idea, taking it one step further; namely, we investigate the use of features derived directly from the image data of the figures (as opposed to just from the text of the figure captions) for biomedical document categorization. It is intuitively clear that image and text data, especially in scientific documents, tend to complement each other. Moreover, psychological studies on the contribution of multimodal data (image, animation, text) to effective understanding in human readers, confirm the efficacy of the combination of image and text for improving the processing and understanding of information by humans, compared with the unimodal form (i.e. either text or image data alone) (Mayer and Moreno, 2002). We report here a first experiment, introducing image features into the text categorization process, and show preliminary results in applying it to a subset of the TREC Genomics data.

*To whom correspondence should be addressed.

Notably, image-based categorization of documents is an established research field (Chen and Blostein, 2006). It is applied in diverse areas ranging from digital library construction and document image retrieval to office automation. Document image classifiers differ vastly in the problems they solve, in their use of training data to construct class models, and in the choice of document features and classification algorithms. There is no single general, adaptable, high-performance image-based classifier, due to the great variety of documents, the diverse criteria used to define document classes, and the ambiguity in the class definition itself. Thus, the specific task at hand needs to be considered when choosing and applying image-based categorization methods in the biomedical domain.

To the best of our knowledge, the use of figure images themselves has not yet been considered for general biomedical document triage and for automated support of biomedical annotation and curation. Perhaps closest to ours is work by Murphy *et al*. (Huang and Murphy, 2004; Murphy *et al*., 2004), which uses image categorization for identifying subcellular localization articles. They provide an excellent in-depth investigation of a specific task: identifying and interpreting a specific type of image that is characteristic of localization experiments. While their extensive work utilizes information extraction from text to help improve image categorization and interpretation, it is not directed at the integration of text and image features for the purpose of document categorization. Moreover, the research focuses on protein subcellular localization and is not generalized to other biomedical categorization tasks.

In this paper, we explore the possibility of using figures for the document triage task in support of biomedical database curation. We describe a first attempt at using image features for biomedical text categorization, as well as at the integration of such features with the more traditional text-data. The next section outlines the methods we apply, while Section 3 describes the data set and demonstrates preliminary results of applying our integrated categorization method. Section 4 concludes and outlines future work.

2 USING FIGURES FOR DOCUMENT TRIAGE

Document triage can be viewed as a binary classification task. The input is a set of full-text documents, and each document is classified as either *positive* (relevant for annotation) or *negative* (irrelevant for annotation). To automate the task, a classifier is trained using a set of labeled training documents, and is then applied to the test documents to predict their class. Our basic idea is to create an image-based vector description for each document in both the training and the test sets. Once a vector description is created, traditional classification methods can be applied to the data. In this paper we focus on the simple naïve Bayes classifier, although more advanced methods are likely to yield improvement. The image-description approach is adapted from work by Duygulu *et al*. (2002) on content-based image retrieval. Duygulu *et al*. segment images into regions, cluster similar regions across the different images into what they call ''blobs'', and thus create and use a small vocabulary of characteristic segments for representing images. Through most of this section, (2.1, 2.2), we describe our image feature extraction and the document representation in terms of image features. The last part of the section (2.3) provides a brief description of a first integrated framework for combining image features and text data for biomedical document classification. Our experiment and results using a subset of the TREC Genomics 2004 data are described in Section 3.

2.1 Document descriptors via image features

As with any supervised text categorization task, the training data consists of documents that have been manually labeled by human curators as *positive* or *negative*. Typically in text categorization, the documents are then represented as weighted vectors of terms or of words. (For reviews see: de Bruijn and Martin, 2002, Shatkay and Feldman, 2003.) In the heart of our approach is the representation of documents *as vectors of image features* rather than of text features[1], which we describe in detail below.

Before delving into the details, in a nutshell the method comprises five main steps: First, figures are *extracted* from the full-text documents. As single figures often display multiple pictures, they are broken in a *segmentation* step into subfigures. These subfigures are then *classified* into several high-level types of images that we have defined. These three steps are shown in Figure 1. Within each class, *clustering* is then applied to refine the grouping of images by specific contents. Each subfigure is assigned an identifier coding its class and its cluster. In the final step, each document is then *represented* as a vector over the space of subfigure-identifiers as features (similar to the vector space over terms or words typically used in text). We discuss these steps in detail below.

a) Figure extraction. This step starts with full-text XML documents. Captions and links to the figures are extracted from the XML format, figure images are downloaded from the publisher's web site. A sample document is shown in Figure 1(i). One of the extracted figures is shown in Figure 1(ii). For the training and tests described here we used a total of about *4,400* figure images, of which *1,900* came from the training and *2,500* from the test documents.

b) Figure segmentation. As evident from Figure 1(ii), each image may consist of several subfigures. Each image is thus segmented into its subfigures using an approach based on connected components analysis (Gonzalez and Woods, 2002). Such analysis is performed on thresholded black-and-white images, where connected components are regions of neighboring foreground pixels. The connectedness is defined based on eight-neighbors of each pixel. Figure 1(iii) demonstrates the results of such segmentation. We note that this is not a fool-proof procedure, and errors are expected to occur. In the data described here, we identified a total of about *26,500* subfigures (*∼11,000* in the training and *∼15,500* in the test set).

c) Subfigure classification. The subfigures identified in step *b* may illustrate various types of data and be organized in a variety of layouts. As pointed out by Murphy *et al*. (2002), there are no uniform standards for figure organization in the scientific literature. As shown in Figure 2, we have identified several prominent types of figures in the scientific literature and use these types for categorizing subfigures. Obviously this ''ontology'' of image types is neither complete nor perfect, but has proven to be a useful first step for the limited scope in which it is used here.

Subfigure classification forms the basis for creating labels that are later used to represent image features in each figure. Currently, at

[1]We note that for combining text and figures we do use *both* text and image features.

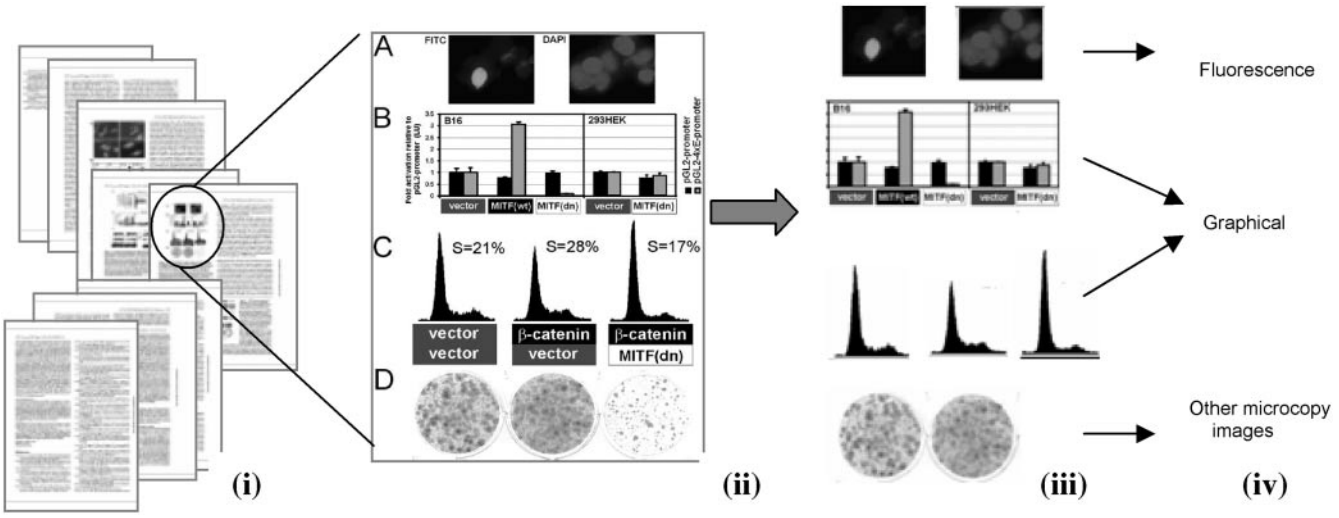


Fig. 1. (i) A sample input document with PubMed Identifier 12235125 (Widlund *et al*., 2002). (Figures reproduced with permission of the Rockefeller Univsrsity press.) The document has nine pages and *six* figures. (ii) Extract all the figures from the document and save as image formats, such as JPEG or GIF. One of the extracted figures is shown enlarged. (Corresponds to step *a* below.) (iii) Figure segmentation based on Connected Components analysis. Subfigures are extracted from each figure. Connected components whose bounding box areas are too small are discarded since they are most likely characters used to label figures. The example document has a total of 39 subfigures. (Step *b* below.) (iv) Subfigure classification using a hierarchical scheme as defined in Figure 2. (Step *c* below.)

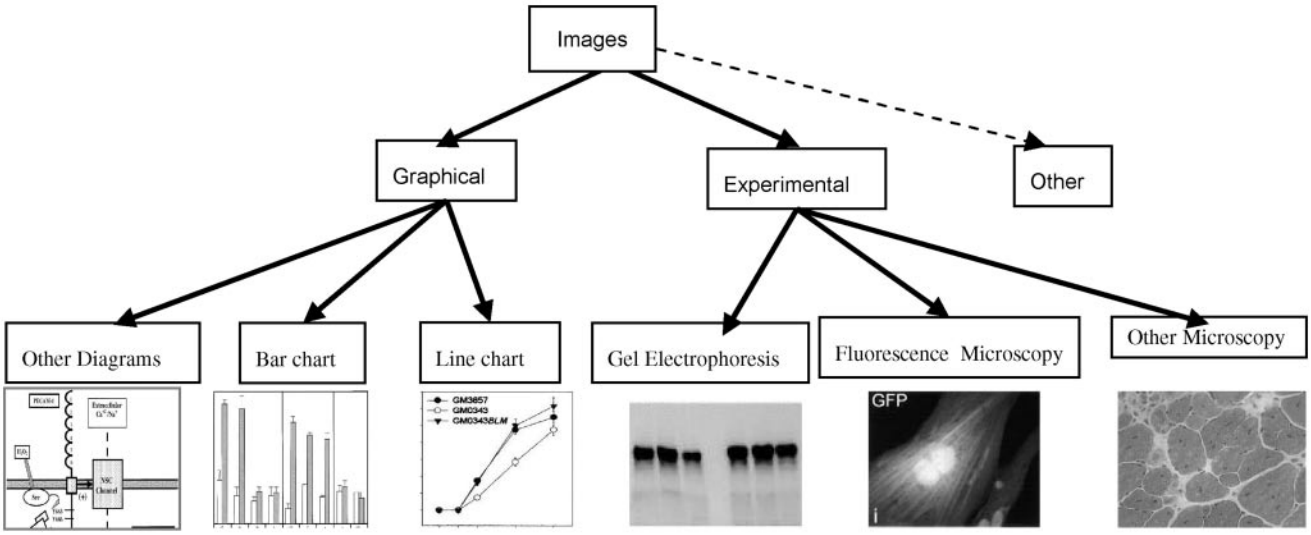


Fig. 2. The hierarchical image classification scheme for subfigures. A sample image is shown for each class. At the top level, images are classified into *Graphical* and *Experimental* images. Other types of images found in publications include photographs such as pictures of mice, author images, etc. In our current work, we manually pre-filter the extracted subfigures to remove such *Other* images. At the second level, *Experimental* images are classified into *Fluorescence Microscopy*, *Gel Electrophoresis*, and *Other Microscopy* images. *Graphical* images are classified into *Line Charts, Bar Charts*, and *Other Diagrams*. In our experiments, *Graphical* images are not further classified. We focus on classification of *Experimental* images into *Gel Electrophoresis*, *Fluorescence Microscopy*, and *Other Microscopy* images.

the first level, images are classified into *Graphical*, *Experimental* and *Other* classes. For the *Experimental* class, we currently define only three subclasses: *Fluorescence Microscopy*, *Gel Electrophoresis*, and *Other Microscopy*. These three subclasses are visually distinct and correspond to clearly different experimental settings. Obviously, more classes should be defined to accommodate other types of experimental imaging. *Graphical* images can also be partitioned into subtypes. For instance: *Line Chart*, *Bar Chart* and *Other Diagrams*. However, in the experiments described here graphical images are *not* further partitioned.

In order to train a classifier to categorize subfigures under this classification scheme, we manually labeled a few hundred subfigures in each class (500 *Graphical* subfigures, 500 *Fluorescence Microscopy*, 300 *Gel Electrophoresis*, and 300 *Other Microscopy*). We use two Support Vector Machine (SVM) classifiers: one at the root level to classify the images into *Graphical* vs.

Experimental images, and the other at the second level of the classification hierarchy to further classify *Experimental* images into one of the three subclasses. Thus, every subfigure is assigned one of four class labels: *Graphical*, *Fluorescence Microscopy*, *Gel Electrophoresis*, or *Other Microscopy*. Examples of subfigure classification results are shown in Figure 1(iv). Using a stratified 10-fold cross validation, the first level classifier for separating *Graphical* from *Experimental* subfigures demonstrates about *95%* accuracy, while the second classifier that separates the three types of experimental subfigures demonstrates a level of *93%* accuracy. Note that this is *not* the ultimate categorization task discussed in this paper; rather, it is a preprocessing step used towards representing images that appear in scientific papers.

To facilitate classification by SVM, subfigures must be represented as feature vectors. The following 46 features are used for representing subfigures in this stage:

- *Statistics based on gray-level histograms*. The histograms represent the distribution of pixels in the subfigures according to their gray-level. Four statistics are derived from the histogram: the first three moments (mean, variance, and skewness) as well as the entropy of the gray-scale distribution (Gonzalez and Woods, 2002).
- *Haralick's texture-features* (Haralick *et al*., 1973), based on the co-occurrence of pixels within the subfigure. The co-occurrence matrix provides information about co-occurring pixels of specific values, orientation and distance. Six features are derived from the matrix including, among others, *contrast* (variation in gray level), *correlation* (likelihood of co-occurrence for specified pixel pairs), and *homogeneity* (formally described as *Inverse Difference Moment*).
- *Edge direction histogram* (Jain and Vailaya, 1998), originally used for shape-based retrieval. Edges are detected in the subfigure, using Canny's edge detector (Canny, 1986). A histogram which bins together edges sharing a similar direction is then formed. Our implementation uses a bin granularity of 10°, resulting in a histogram of 36 bins. The bin sizes (i.e. the number of edges in each of the bins) are used as features.

The image feature vectors are normalized before classifying them. Classification is done using Weka's (Witten and Frank, 2005) implementation of Support Vector Machines, with the radial basis function kernel.

d) Subfigure clustering into finer groups. In the previous step subfigures were classified into one of four coarsely-defined classes. In the relatively small training set (256 documents) described here alone there were about 11,000 subfigures. As it was expected that the four broad manually defined classes, while intuitively clear, are unlikely to provide sufficient discrimination among thousands of subfigures, we use unsupervised clustering to refine the grouping of similar and related images into tight subsets. Since the number of subfigures assigned to the *Fluorescence Microscopy* class is about 4 times larger than the number of subfigures assigned to each of the other two classes, the Fluorescence Microscopy class is sub-clustered into 20 clusters, while the other classes are sub-clustered into 10 clusters each. Clearly, a different number of clusters may be used, and may yield different results. We have chosen the current numbers based on the total size of the image set used here, the total number of sub-figures stemming from it, and based on several experimental runs. We expect to test more methodically in future studies how the number of clusters affects the classification performance. While this is an interesting point whenever clustering is concerned, it is not a central issue for the work presented here.

```
graphics graphics graphics F19 graphics
graphics E2 F17 F9 F19 F16 graphics
graphics graphics graphics G6 G7 graphics
G1 graphics G3 graphics F17 G0 graphics
graphics graphics graphics E7 F6 G6 E5
graphics E1 graphics E5 G1 G4 graphics
```

Fig. 3. The document shown in Figure 1(i), represented using only subfigure identifier terms.

The clustering step groups together images with similar characteristics. In this study, we use the simple k-means algorithm, as implemented in Weka (Witten and Frank, 2005). The features considered are the same ones used for the subfigure classification described in step *c* above. As this is a first study on the use of images for biomedical text categorization, we have not yet explored the range of possibilities for representation, classification and clustering, and expect to do so in the future. A discussion of the variety of methods for document image classification techniques is given in a previous survey (Chen and Blostein, 2006).

To summarize this stage, subfigures within each of the four classes that were formed in step *c* are clustered into finer groups. The clustering results are used to assign a cluster label to each subfigure, which together with the class label serve to characterize each subfigure in every document.

e) Document representation as an image-based feature vector. In steps *c* and *d* each subfigure has been assigned both a class name and a cluster number. Combined, this information forms a label characterizing each subfigure in terms of its class and cluster. For example, the top left subfigure in Figure 1(iii) is assigned the label *F17*, where *F* stands for *Fluorescence Microscopy* and *17* stands for *cluster 17* among the 20 clusters of *Fluorescence Microscopy* subfigures. The labels of all the subfigures in each document are taken as new kinds of terms used to represent each document based only on its image features. A feature vector is then constructed from the description, similar to the way weighted term vectors are built from text. For example, the description of the document shown in Figure 1(i) is shown in Figure 3 (before vectorization and term weighting is performed). In this description, G represents *Gel Electrophoresis*, F represents *Fluorescence Microscopy* and E represents *Other Microscopy*, while ''graphics'' denotes subfigures that are non-experimental *Graphical* images. This image description was created by concatenating the labels of 39 subfigures, comprising the six figures in the whole article.

The corresponding vector representation under a simple term-frequency weighting scheme is shown in Figure 4. This is a 41-dimensional vector, as there are 10 *Gel Electrophoresis* clusters, 20 *Fluorescence Microscopy* clusters, 10 *Other Microscopy* clusters, and a single *Graphical* class that is not subclustered. In this case each number in the vector represents the number of times the respective feature occurred in the representation shown in Figure 3.

2.2 Image-based classification with naïve Bayes

Given the image-based description created in step *e* above, each document is further converted into an *n*-dimensional feature vector,

E0	E1	E2	…	E5	E6	E7	E8	E9	F0	…	F6	…	F9	…	F16	F17	F18	F19	G0	G1	G2	G3	G4	G5	G6	G7	G8	G9	graphics
<0	1	1	…	2	0	1	0	0	0	…	1	…	1	…	1	2	0	2	1	2	0	1	1	0	2	1	0	0	19 >

Fig. 4. The vector representation for the document shown in Figure 1(i) and Figure 3, using term-frequency weighting. The feature labels are listed above their weights. In the weight vector, '. . .' indicates a sequence of consecutive 0's.

where n is the total number of distinct image-based terms (where a *term* is a descriptor such as ''graphics'' or ''*E7*'' above). For each article, every such term is weighted according to its frequency in the article, using MALLET's (McCallum, 2002) default weighting scheme.

Once the feature vectors are formed, we build a naïve Bayes Classifier using all the training documents, to distinguish *positive* articles (relevant for curation) from *negative* ones (irrelevant for curation). Naïve Bayes is a simple and popular classification method; given its simplicity and ease of implementation, it performs well in practice (Mitchell, 1997). The naïve Bayes classifier is built by obtaining statistics from the set of labeled training data. A document D, represented by its feature vector $(d_1, \ldots, d_n)$, where in our case d_i is the weight of the i^{th} subfigure-identifier term, is assigned to the class C that maximizes the likelihood: $\Pr(D \mid C) = \prod_{i=1}^{n} \Pr(d_i \mid C)$.

Expressing the conditional probability $\Pr(D \mid C)$ as a product of simpler probabilities is based on the (naïve) assumption of conditional independence among the features, given the class. We use the MALLET toolkit (McCallum, 2002) for feature vector creation and for the naïve Bayes classification of documents. We note that although MALLET was originally built for text processing and categorization, we use here image-derived features (as shown in Figure 3) rather than text features as input to MALLET.

The representation and training steps given above, when applied to the training data, result in clusters and classifiers for subfigures (steps *c, d* above), which allow each document to be represented based on its image contents (steps *a-e* above). More importantly they yield a naïve Bayes classifier for categorizing documents, using their image-based representation. Given a *new input document*, we classify it by executing the following procedure: First, the document goes through steps *a-c*, namely, its figures are extracted, segmented and its subfigures classified, in a way similar to the preprocessing applied to the training data. Then each subfigure is assigned the cluster label of its nearest neighbor in the training set, using the results of training step *d*. An image-based description is created containing a list of labels of all the subfigures in the document, similar to training step *e*. Then a feature vector is computed and fed into the naïve Bayes classifier described above. This classifies the input document as *positive* or *negative* based on its relevance to the curation task at hand.

2.3 Integration with a simple text classifier

As a first attempt at integration of text data with image features, we use the simplest and most widely used and readily available text for biomedical documents, namely only the title and the abstract of the articles as they appear in PubMed. The titles and abstracts of all the articles contained in both the training and the test set were tokenized to obtain a dictionary of terms consisting of single words (unigrams) and pairs of consecutive words (bigrams), where words were stemmed using the Porter stemmer (Porter, 1997) and standard stop-words removed. Rare terms (appearing only in a single document) as well as very frequent ones (occurring in more than 10% of the documents) were also removed. The remaining terms, along with their frequencies within each of the documents were used to create, for each article, a representation similar to the one shown in Figures 3 and 4, only in this case the features are the actual text-terms. The abstracts of articles in the training set were then used, as described in Section 2.2 to train a naïve Bayes classifier using the MALLET toolkit (McCallum, 2002). We note that both the preprocessing and the classification schemes here are basic ones, and will be extended in the very near future.

The integration scheme for combining the text and the image classifiers consists of a simple OR combination, where a document is considered as relevant for the triage task if either the text-based classifier or the image-based classifier identified it as relevant. This strategy is based on the observation that the triage task stressed the importance of retrieving as many relevant documents as possible, even at the cost of drawing in false-positives (more detail is given in the next section).

3 EXPERIMENTS AND RESULTS

3.1 Experimental setting

We test our method on a subset of the data that was used for the categorization task in the TREC Genomics Track 2004 (Hersh *et al*., 2005), and specifically focus on the *triage* task. The *triage* task aimed to classify documents as *relevant* or *irrelevant* for supporting GO annotation by curators for the Mouse Genome Informatics (MGI) resource at the Jackson labs. The original dataset consisted of full-text articles from three journals: *The Journal of Biological Chemistry (JBC), The Journal of Cell Biology (JCB)*, and *The Proceedings of the National Academy of Science* (*PNAS*), over the period of two years, 2002 and 2003. The 2002 articles (a total of 5,837) were designated as the training set for the task, while those from 2003 (6,043 such articles) as the test set. The true triage decisions were provided by MGI.

In the experiments described here, we use only documents from the *Journal of Cell Biology (JCB)* as provided in TREC Genomics 2004. It is important to note that image data was *not* included in the TREC data set. Given the non-trivial time and effort needed to obtain the image data, download and process it, and given that this is the first study to use biomedical image data for biomedical literature categorization, we wanted to first validate the feasibility of the task and establish a well-defined pipeline, before embarking on the more ambitious task of utilizing the full amount of available data. The distribution of training and test data used here is shown in Table 1.

We train a classifier based on the images from the 256 training documents, and test it on the 359 test documents. A simple text-based classifier is trained on just the abstracts and titles of the same set used for training the image-based classifier, and tested on the

Table 1. The distribution of positive and negative documents in the training and test data sets

	Positive documents	Negative documents	Total figures extracted	Total subfigures extracted	Total documents
Training JCB'02	26	230	1,881	10,920	256
Test JCB'03	34	325	2,549	15,549	359

abstracts and the titles of the same test set as used in the image case. Finally, an integrated classifier assigns a document as *relevant for curation* if *either* of the two first classifiers tagged it as *relevant*. To evaluate our results, we use the same metrics used to assess the triage subtask in the TREC 2004 Genomics track. The primary evaluation metric for the triage subtask, as defined by Hersh *et al*. (2005), was the normalized *Utility* value, defined as:

$$U_{norm} = \frac{(20 \cdot TP) - FP}{20 \cdot Pos}$$

In this formulation, *TP* is the number of true positives (documents that were relevant for curation according to MGI, and identified by the classifier as relevant), *FP* is the number of false positives (documents identified by the classifier as relevant, but not considered as such by MGI), and *Pos* is the total number of articles that are relevant according to MGI. The constant *20* was introduced by Hersh *et al*., and serves to bias the evaluation to favor high recall (that is, including as many positive examples as possible). It reflects the notion that missing a relevant document that should be curated is considered much more costly than including an irrelevant document. Hersh *et al*. (2005) indicated that the ideal approach for determining this constant would involve interviewing MGI curators and formally determining utility, but they used a simplified approximation for the time being. Other measures include the standard *precision*, *recall,* and *F-score* (combining recall and precision). The formulae for these last three measures are as follows, where we again use the abbreviations *TP* (True Positive), *FP* (False Positive), *FN* (False Negative):

$$\text{Precision: } \frac{TP}{TP + FP} \quad \text{Recall: } \frac{TP}{TP + FN}$$

$$\text{F-score: } \frac{2 \cdot recall \cdot precision}{recall + precision}$$

3.2 Results

Table 2 summarizes our results from training and testing over the JCB dataset (as shown in Table 1).

It is important to note that while our results are in the same utility range as that obtained by TREC — and the combined utility of the integrated system may look even higher than that achieved by the average TREC run — our numbers (the top three rows) *do not* compare directly with the TREC 2004 Triage results (the bottom row), because we use only a *subset* of the TREC training and test documents. The bottom row is provided not for comparing our classifiers with those of TREC, but rather to provide a ''ballpark'' range for what one may expect to see in such results, and to demonstrate that our results fall in this range. Meaningful comparative analysis can only be made among the numbers presented in the top three rows.

Table 2. Classification results, using the evaluation metrics described by Hersh *et al*. (2005). Average results from the TREC 2004 Triage runs, taken from Table 6 of Hersh *et al.'s* report (2005), are shown for an informal comparison. Due to the efforts involved in obtaining figure images, we only used a fraction of the test and training documents used in the TREC Triage task, as shown in Table 1. Our testing used 34 positive and 325 negative documents, whereas the TREC 2004 Triage testing used 420 positive and 5,623 negative documents

	Utility	Precision	Recall	F-score
Image-features system	0.307	0.279	0.353	0.312
Simple text classifier	0.315	0.647	0.323	0.431
Integrated	**0.446**	0.315	0.5	0.386
Avg. of 59 runs in TREC'04 triage task	0.330	0.138	0.519	0.195

All 59 of the TREC 2004 Triage runs were based on full-text documents[2], including figure captions, but not including any analysis of figure images. In contrast, our results for the image-based classifier makes no use of text and uses *only image data*, while the text-based classifier uses only the title and the abstracts of the documents with no other information. The combined classifier takes only the output of these two classifiers to make a categorization decision. As shown in Table 2, our results are well within the numerical range of the average results in TREC 2004 runs. This is encouraging, indicating that even with very simple features the image-based classifier can achieve a reasonable level of performance.

Most importantly, we note that the integration of the image classifier and our simple text classifier significantly improves upon the *utility* obtained by each of the individual classifiers alone. As explained in the previous section, this integration is performed by assigning the tag *relevant*, to a document if any of the two first classifiers categorized it as *relevant*. The fact that this strategy improves recall, (and in-turn utility), indicates that the two original classifiers are not strongly dependent, and use different criteria to reach their conclusions. This is an important observation, given that combining classifiers relies on the idea that an ensemble of classifiers improves performance with respect to its individual components if these components are mostly independent of each other (Sebastiani, 2002, Tumer and Ghosh, 1996). These preliminary results and the nature of both images and text in scientific documents indicate that the combination of figure and text analysis has the potential to yield good results. We expect that image data, which

[2]Notably, not all 59 runs took advantage of the full text; some participants utilized only parts of it, such as abstract, title or MeSH terms.

is a condensed form of information specific to certain types of scientific discussions, will complement the information conveyed in the natural-language text.

4 DISCUSSION AND FUTURE WORK

The research presented here is a first exploration of the possibility of using image data in support of document categorization in the biomedical domain. We note that the idea of using figures for the end goal of text classification is novel and has not been applied yet even in the general context of text categorization (i.e. outside the biomedical domain). In our current work we used a rather small data set, simple methods for segmentation, classification and clustering of subfigures, as well as a very basic text classification and integration strategy. The results of even this simple approach are encouraging and suggest that image data has much to offer in support of biomedical text categorization. A refinement of all these steps is expected to improve the end result. An important immediate step is the application of both the current and the refined methods to the full data set, and specifically to the TREC'05 categorization tasks[3]. Experiments with the GO and Allele categorization tasks of TREC'05 (Hersh *et al*., 2006) over the JCB subset, using appropriately adapted utility scaling measures, yield results similar to the ones shown in Table 2. We are already running the system on the complete data set, and are currently experimenting with categorization, clustering and feature selection strategies that are appropriate for this much larger and heterogeneous data set.

Experiments with other classifiers, aside from the naïve Bayes, as well as the application of more advanced text-categorization and the use of text from captions and other parts of the document, are natural and essential directions we are currently pursuing. Another important next step is the study of the complementary role of text and image data in biomedical text categorization. We are interested in combining the analysis of text, ontology, and figures for document triage and annotation tasks.

In our future research, we shall investigate how human curators use figures in judging whether a document supports annotation, and how figures are used during the annotation process. Observing how humans handle the task will provide further ideas on how to automate (parts of) it. As noted in the introduction, Mayer and Moreno (2002) examined the role of text and diagrams in understanding scientific literature and assessed whether visual information improves recall and problem-solving skills in human readers. They observe that properly organized multimodal presentations improve human performance in understanding the presented material. Given the condensed and informative nature of scientific images, and the rapidity in which humans perceive, process, and reach decisions based on such visual cues, we expect images in biomedical text to provide an invaluable support for categorization and mining of such text. We view text- and image- based document categorization as highly complementary, rather than competing approaches.

Our current results, along with these observations and the already accepted notion that database curators strongly rely on image data in articles to support their decision, strengthen our hypothesis that utilizing images can improve document categorization. Combining image analysis with text analysis is thus expected to help resolve ambiguity and improve the effectiveness of literature mining. The preliminary results presented here, from categorizing biomedical documents using both text and image data, further demonstrate and support this idea.

There are several challenges when applying document image analysis techniques for biomedical literature mining. In contrast to the millions of abstracts in MEDLINE, the number of full-text documents is still limited. Easy-to-use electronic versions (e.g. articles in XML format), with separately accessible figures and text are available for some papers, but not for all. For other cases (e.g. articles in PDF or image format), preprocessing has to be performed to separate text and figures, and to associate figures with figure captions. This preprocessing is difficult and error prone. Moreover, training and test data based on curation decisions is not available for individual images, but only for complete documents. We are actively pursuing ways to obtain labeled images that have been used by curators to determine the relevance/irrelevance of documents. We believe that having access to such data would form a major step forward in training classifiers that utilize image data for text categorization.

[3]TREC'04 participants noted that the data for the 2004 Triage task had some limitations. Additional categorization tasks, and different utility scaling measures were defined for TREC'05.

ACKNOWLEDGEMENTS

We thank Scott Brady for his kind help. We gratefully acknowledge the financial support provided by NSERC—Canada's Natural Sciences and Engineering Research Council, CFI—the Canadian Foundation for Innovation, and by the Xerox Foundation.

REFERENCES

B. de Bruijn and J. Martin. (2002). Getting to the (c)ore of knowledge: mining biomedical literature. *Int. Journal of Medical Informatics* 67(1-3). pp. 7-18.

J. Canny. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6). pp. 679-698.

N. Chen and D. Blostein. (2006). A Survey of Document Image Classification: Problem Statement, Classifier Architecture and Performance Evaluation. *International Journal of Document Analysis & Recognition*. (In Press).

K. Darwish and A. Madkour. (2005). The GUC goes to TREC 2004: Using whole or partial documents for retrieval and classification in the Genomics Track. *Proc of TREC 2004*. NIST Special Publication. pp. 362-369.

P. Duygulu, K. Barnard, N. de Freitas and D. Forsyth. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *In Proc. of the Seventh European Conference on Computer Vision*. pp. 97–112.

R.C. Gonzalez and R.E. Woods. (2002). *Digital Image Processing*. Prentice-Hall.

R.M. Haralick, K. Shanmugam, and I. Dinstein. (1973). Texture features for image classification. *IEEE Trans. On Systems, Man and Cybernetics*. SMC-3(6). pp. 610-621.

W.R. Hersh, R.T. Bhuptiraju, L. Ross, P. Johnson, A.M. Cohen, D.F. Kraemer. (2005). TREC 2004 Genomics Track overview. *Proc. of TREC 2004*, NIST Special Publication. pp. 132-141.

W.R. Hersh, A. Cohen, J. Yang, R.T. Bhuptiraju, P. Roberts, M. Hearst. (2006). TREC 2005 Genomics Track overview. *Proc. of TREC 2005*, NIST Special Publication. pp. 14-25.

L. Hirschman, A. S. Yeh, C. Blaschke and A. Valencia. (2005). Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*. 6(Suppl 1):S1. pp. 1-10.

K. Huang & R.F. Murphy. (2004). From quantitative microscopy to automated image understanding. *Journal of Biomedical Optics*. 9(5). pp. 893-912.

A.K. Jain and A. Vailaya. (1998). Shape-based retrieval: a case study with trademark image databases. *Pattern Recognition*, 31(9). pp. 1369-1390.

R.E. Mayer and R. Moreno. (2002). Aids to computer-based multimedia and learning. *Learning and Instruction*, 12. pp. 107-119.

A. K. McCallum. (2002). MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu.

T. Mitchell. (1997). *Machine Learning*. McGraw-Hill.

R.F. Murphy, Z. Kou, J. Hua, M. Joffe, W.W. Cohen. (2004). Extracting and structuring subcellular location information from on-line journal articles: the Subcellular Location Image Finder. *Proc. of the IASTED International Conference on Knowledge Sharing and Collaborative Engineering (KSCE-2004)*. SLIF server web site: *http://goblin.cbi.cmu.edu:8080*

M.F. Porter. An Algorithm for Suffix Stripping (1997, Reprint). In *Readings in Information Retrieval*. Morgan Kaufmann.

Y. Regev, M. Finkelstein-Landau, R. Feldman, M. Gorodetsky, X. Zheng, S. Levy, R. Charlab, C. Lawrence, R.A. Lippert, Q. Zhang, H.Shatkay. (2002). Rule-based extraction of experimental evidence in the biomedical domain—the KDD Cup (Task 1). *SIGKDD Explorations* 4(2). pp. 90-92.

F. Sebastiani. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys* 34(1). pp. 1-47.

H. Shatkay and R. Feldman. (2003) Mining the biomedical literature in the genomic era: an overview. *Journal of Computational Biology,* 10(6). pp. 821-855.

K. Tumer and J. Ghosh. (1996). Error Correlation and Error Reduction in Ensemble Classifiers. *Connection Sci.* 8(3-4). pp. 385-403.

H.R. Widlund, M.A. Horstmann, E.R. Price, et al. (2002). Beta-catenin-induced melanoma growth requires the downstream target Microphthalmia-associated transcription factor. *Journal of Cell Biology*. 158(6). pp. 1079-87.

I. H. Witten and E. Frank. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann. (Describes Weka: The Waikato Environment for Knowledge Analysis. http://www.cs.waikato.ac.nz/ml/weka.)

A.S. Yeh, L. Hirschman, A.A. Morgan. (2003) Evaluation of text data mining for database curation: Lessons learned from the KDD Challenge Cup. *Bioinformatics*, 19. pp. i331-i339

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e454–e463
doi:10.1093/bioinformatics/btl227

On counting position weight matrix matches in a sequence, with application to discriminative motif finding

Saurabh Sinha

Department of Computer Science, University of Illinois, Urbana-Champaign, 201 N. Goodwin Ave, Urbana, IL 61801, USA

ABSTRACT

Motivation and Results: The position weight matrix (PWM) is a popular method to model transcription factor binding sites. A fundamental problem in cis-regulatory analysis is to "count" the occurrences of a PWM in a DNA sequence. We propose a novel probabilistic score to solve this problem of counting PWM occurrences. The proposed score has two important properties: (1) It gives appropriate weights to both strong and weak occurrences of the PWM, without using thresholds. (2) For any given PWM, this score can be computed while allowing for occurrences of other, *a priori* known PWMs, in a statistically sound framework. Additionally, the score is efficiently differentiable with respect to the PWM parameters, which has important consequences for designing search algorithms.

The second problem we address is to find, *ab initio*, PWMs that have high counts in one set of sequences, and low counts in another. We develop a novel algorithm to solve this "discriminative motif-finding problem", using the proposed score for counting a PWM in the sequences. The algorithm is a local search technique that exploits derivative information on an objective function to enhance speed and performance. It is extensively tested on synthetic data, and shown to perform better than other discriminative as well as non-discriminative PWM finding algorithms. It is then applied to cis-regulatory modules involved in development of the fruitfly embryo, to elicit known and novel motifs. We finally use the algorithm on genes predictive of social behavior in the honey bee, and find interesting motifs.

Availability: The program is available upon request from the author.

Contact: sinhas@cs.uiuc.edu

1 INTRODUCTION

The study of transcriptional regulation is a pervasive topic in bioinformatics, due to growing realization of the role it plays in all cellular processes, and in the evolution of organismal novelty. A crucial step in such studies is to identify transcription factor binding sites that mediate the regulation of genes. This has proved to be a challenging computational task, largely due to the high variability and short length of binding sites of any given transcription factor (Tompa *et al*., 2005). To model this variability, the position weight matrix (PWM) has emerged as a probabilistic construct of popular choice. A PWM specifies the frequency distribution of nucleotides at each position of the binding sites, and is considered to be related to the energy of binding of the transcription factor to the DNA (Stormo and Fields, 1998). The motif finding problem is to find a PWM representing binding sites of an unknown transcription factor, *ab initio* from sequence data.

A particular variant of this problem, the discriminative motifs problem, is *to find PWMs that are present in one set of sequences (the positive set) and absent in another set of sequences (the negative set)*. There is an abundance of data sets that pose this problem. For example, the segmentation pathway in fruitfly comprises genes that are expressed in certain spatial domains in the embryo, and genes that are not expressed in those domains. Such information is easily available (Tomancak *et al*., 2002), and this spatial partitioning of genes is known to be under transcriptional control (Schroeder *et al*., 2004). In general, any gene expression data set may be mined to obtain positive and negative sets of genes (and their promoters/enhancers), thereby presenting a typical instance of the discriminative motifs problem. Compared to the traditional motif-finding approach, where the positive set of sequences is contrasted with a large ''background'' sequence set, discriminative motif-finding presents much cleaner and more informative negative sequence data to contrast with—the promoters that *do not contain* the desired motif. However, in contrast to the vast body of literature on the motif finding problem, the amount of research done on this useful variant of the problem, especially for the PWM model, is very little.

To solve this problem, the first question to address is: How do we ''count the occurrences'' of a PWM in a sequence? No satisfactory answer has emerged for this problem, despite the wide acceptance of PWMs as a motif model. Let us see some of the implications of the question, and pitfalls of some possible answers.

(1) Counting is different from simply asking if a PWM occurs in the sequence (a ''yes''/''no'' question), and it is inadequate to simply report the quality of the best match to the PWM in the sequence. This is especially relevant for cis-regulatory sequences where the number (and strengths) of binding sites determines function (Schroeder *et al*., 2004).

(2) An ''occurrence'' of a PWM has been traditionally defined by the ''sampling probability'', which is the probability of sampling a given k-mer from the probability distribution induced by a k-length PWM. The naïve approach then is to count all k-length substrings that have sampling probability above a certain threshold, as occurrences of the PWM. The problem with this approach is that one does not differentiate

strong (high sampling probability) occurrences from weaker occurrences, as long as they are above the threshold.

(3) In many motif-finding applications today, there is prior knowledge of one or more motifs that are relevant to the data set being analyzed. It is thus extremely useful to be able to count motif occurrences while factoring in the presence of these known motifs. The standard heuristic used for this is to mask out (partly or completely) the strong matches to the known motif(s) in the sequences, as a pre-processing step. This heuristic may be problematic if a match to a PWM either *overlaps with* or *is* a match to another (known) PWM.

We propose a novel probabilistic score, called the ''w-score'', to quantify the total number of occurrences of a PWM in a sequence, while handling strong and weak occurrences appropriately. (This addresses points 1 and 2 above.) Intuitively, the w-score may be understood as the average number of times the PWM is ''planted'' by a probabilistic model generating the sequence, the average being over all possible ways to generate the sequence as a concatenation of PWM and background sites. Since the score is based on a probabilistic model for sequence generation, any known motif(s) may be incorporated into the model as prior knowledge, and the score of a PWM computed in the context of the known PWMs. (This addresses point 3 above.) The known motifs may be those that were computationally discovered in previous executions of the program.

Having addressed the problem of *counting* occurrences, the next question is: How to find PWMs (motifs) with high counts (w-scores) in the positive set of sequences and low counts in the negative set? One has to define a ''discrimination score'', which captures how different a PWM's counts are in the two sets, and devise an algorithm that maximizes such a score over the space of PWMs. We implement two different discrimination scores—one that directly compares the average counts (w-scores) in the two sets of sequences, and one that models the problem as a classification task based on the counts. We propose a novel hill-climbing algorithm that exploits derivative information on the discrimination score to guide the search. Certain algorithmic choices, explained in Section 3.3, make the algorithm less susceptible to local optima as opposed to a conjugate gradients search. (See Section 4.) It is worth noting that the proposed PWM-finding framework can be trivially modified to optimize other objective functions (not necessarily a discrimination score), e.g., correlation of motif counts with gene expression data.

The main novel contributions of this work are:

(1) A probabilistic model-based score to count PWMs while accounting for number and strengths of occurrences and incorporating any known motif(s), within a statistically sound framework.

(2) An algorithm for the discriminative motif-finding problem, that is based on the new PWM-counting score, and which is empirically shown to be resilient to local optima.

2 PREVIOUS WORK

The work of Jensen and Liu, 2004 proposed a Bayesian score to be used by a PWM search algorithm, though not in a discriminative setting. This score *evaluates a specific set of sites* assumed to be occurrences of an unknown PWM, and their algorithm BioOptimizer searches for the highest scoring set of sites. In contrast, the w-score evaluates a PWM by considering *all possible* sets of occurrences, weighing each such set by its likelihood. In other words, the w-score ''sums away'' the hidden variables representing motif occurrence positions. (This is done at the expense of added but manageable time complexity.) Segal *et al*., 2003 counted occurrences (of a *k*-length PWM) by summing the sampling probability (defined above) of all *k*-mers in the sequence, and deployed a conjugate gradients search algorithm to find discriminative motifs. However, their underlying probabilistic model allows exactly one motif occurrence in a sequence, and their motif-counting score is therefore different from the w-score. Moreover, the principled incorporation of known PWMs into the score is an important advantage of our approach over that of (Segal *et al*., 2003), and this is demonstrated experimentally in Section 4. The work of Xing *et al*., 2003 (LOGOS) provides a general framework for sequence analysis with multiple motifs. It takes a Bayesian approach to motif detection, allows for prior distributions on PWMs, and in fact allows for a more general motif model than PWMs. LOGOS uses a Hidden Markov model (HMM) for distribution of motif occurrences in a sequence, which is identical to the model used in defining the w-score. As such, both LOGOS and our approach can learn motifs while allowing for multiple motifs and handling the statistical dependency of overlapping motif occurrences. However, in contrast to the Bayesian approach of LOGOS, we use an expectation (derived from the same model) as the motif score, and this choice is crucial in how the discriminative motif-finding problem is solved. (The LOGOS framework does not allow finding discriminative motifs.) The DME program of Smith *et al*., 2005 finds PWMs that are most overrepresented in one set of sequences relative to another set, and uses an enumerative search of a discrete PWM space with a specific lower bound on information content of the PWM. The enumerative search affords nice guarantees on the optimization procedure, but this program, unlike our approach, does not allow incorporating *a priori* known motifs during the search. (It discovers additional motifs by erasing the predicted occurrences of the currently predicted motifs.) Finally, note that the commonly used ''relative entropy'' score (Lawrence *et al*., 1993; Stormo and Fields, 1998) of a PWM measures the specificity of the PWM itself, and does not ''count'' its occurrences in sequences.

The discriminative motif-finding problem was also addressed earlier in Sinha, 2003, Sumazin *et al*., 2005 and Takusagawa and Gifford, 2004, among others. The motif model assumed was the ''consensus string'' model, for which the motif counting problem is trivially solved. However, this is a less sensitive motif model than the PWM, and arguably less realistic for complex transcriptional systems.

3 METHODS

3.1 The w-score

This is a score to represent the number and strength of occurrences of a PWM in a given sequence. We go through an informal description first, and a formal definition will follow. Consider the sequence illustrated in Figure 1a that has substrings (''sites'') A, B, and C matching one of two motifs M1 and M2 as shown in Fig. 1b—overlapping sites A and B are matches to M1 and M2 respectively; site C matches M1. Fig. 1c shows six different ''configurations'' for the sequence. A configuration labels non-overlapping sites as

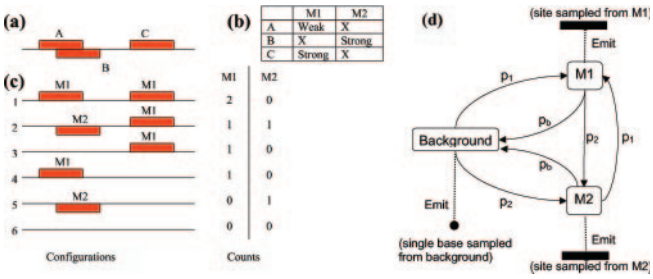


Fig. 1. The w-score. (a) Sites in a sequence (b) Matches between sites and motifs. X means ''no match''. (c) All possible configurations of the sequence with sites assigned to motifs, and count of each motif in each configuration. (d) Generative model (HMM) used to define w-score.

matches to particular motifs. The probability of a configuration depends on the strengths of these (site, motif) matches. For instance, configuration (1) has lower probability than (2) because the (A, M1) match is weak and the (B, M2) match is strong. We can count the occurrences of a motif, say M1, in each configuration, and compute an average of this count, over all configurations weighted by their probability. Roughly speaking, this average count is the ''w-score'' of M1. Note that configurations with weak sites will also contribute (albeit less) to the count. In general, every substring (and not just the three shown) is considered as a potential site for a transcription factor (motif), and will be labeled as a match to that motif, in some configurations. Also note that the set of configurations includes all combinations of (site, motif) matches for *both motifs* M1 and M2. Therefore, taking an average of M1's count over all configurations accounts also for matches to M2 in the sequence, in a natural way.

More formally, the w-score is defined as follows. Suppose we are given a set of PWMs $W = \{w^1, w^2, \ldots w^\kappa\}$, in addition to a ''background'' PWM w^b of length 1. We assume, as in Sinha *et al*., 2003, a stochastic process that generates the sequence from left to right by successively ''planting'' occurrences of PWMs. (Fig. 1d.) At any position, the process chooses to plant a PWM w^i ($i \in \{1, 2, \ldots \kappa, b\}$) with probability p_i. A substring is sampled from the probability distribution defined by w^i (see Appendix), and appended to the sequence generated so far. The process stops when the total length of generated sequence reaches a fixed value L. The p_i, called ''transition probabilities'' are parameters of the model, with $\sum_{i=1}^{\kappa} p_i + p_b = 1$. The sequence of PWMs chosen in successive steps of the process is called a ''parse'' of the sequence, which is exactly the same as a ''configuration'' in the informal description above. Note that the motif location model described here is a zeroth order HMM. (Fig. 1d.)

The model parameters $\Phi = \{p_i\}$ and the PWMs $W \cup \{w^b\}$ associate a well-defined probability $Pr(S, T \mid \Phi)$ with each parse T of a sequence S of length L. This allows us to compute, via Bayes rule, the conditional probability of parse T given the sequence, i.e., $Pr(T \mid S, \Phi)$. We define the w-score of any PWM w^m in the sequence S, with model parameters Φ, as

$$\sigma(w^m, S, \Phi) = \sum_T \chi_m(T) Pr(T \mid S, \Phi) \tag{1}$$

where $\chi_m(T)$ is the number of times w^m occurs in parse T. As we can see, the w-score is the expected number of occurrences of w^m planted while generating S with model parameters Φ. The integral count of w^m in T is weighted by the probability of T given the sequence S. In this sense, the w-score is a natural probabilistic extension of the notion of motif ''count''. The conditional probability $Pr(T \mid S, \Phi)$ is lower for a parse T with weak (low sampling probability) sites than for a parse with equal number of strong sites, implying a lesser contribution made to the w-score by weak sites. Notice that the w-score is defined in terms of a probability distribution induced jointly by all PWMs $W \cup \{w^b\}$, though this dependence is not made explicit in the notation. The w-score can be computed using the Forward-Backward algorithm (Durbin *et al*., 1998) in $O(L \times |W| \times l)$ time, where l is the maximum length of a PWM.

3.2 The discrimination-score

We now define a score to discriminate positive and negative sets of sequences based on w-scores of a PWM w^m in the sequences. Suppose we are given two sets of sequences S^+ and S^-, and the w-score σ_s of w^m for each sequence s, i.e., $\sigma_s = \sigma(w^m, s, \Phi)$. The ''t-score'' is a discrimination score defined as

$$\tau(S^+, S^-, w^m) = \frac{\sum_{s \in S^+} \sigma_s / L_s}{|S^+|} - \frac{\sum_{s \in S^-} \sigma_s / L_s}{|S^-|} \tag{2}$$

($|S|$ for a set S represents its cardinality; L_s for a sequence s represents its length.) The t-score is the difference in mean w-score of the PWM in the two sets of sequences, after normalization of each w-score σ_s by the respective sequence length L_s.

We also defined and implemented an alternative discrimination score, called the ''logistic-score'', described in the Appendix. It is similar in spirit to the treatment in Segal *et al*., 2003 and Hong *et al*., 2005, where a motif score is transformed to a soft class prediction using the logistic function.

3.3 Algorithm

We first provide details of the algorithm, followed by a clearly marked explanation. The algorithm separates the search space from the objective function. Any set of substrings of positive sequences is a candidate motif in the search space, while the objective function is the discrimination score of the unique *PWM constructed from the candidate motif*.

Input : Two sets of sequences S^+ and S^-, integer l_m (desired motif length), background motif w^b and any known motifs.

Parameters : The model parameters $\Phi = \{p_i\}$, integer n (called ''motif cardinality'').

Search space : Any set of n substrings (of length l_m each) of sequences in S^+ is a candidate motif, also called a ''site-set''.

Notation : $\delta(w)$ denotes the discrimination-score to be maximized, e.g., $\delta(w) = \tau(S^+, S^-, w)$ if the t-score is being used. $w_{k\alpha}$ is the weight matrix entry for base α in column k. For any site-set C, we use $W(C)$ to denote the PWM constructed from C in the obvious manner.

Desired Output : Site-set C such that $\delta(W(C))$ is maximized over all C.

Algorithm : Initialize the site-set C to one chosen randomly from the search space. In successive iterations, update C (as described next) to improve $\delta(W(C))$ until no improvement is obtained. Repeat the entire process a fixed number of times, starting from new initial site-sets, and report the site-set with the highest score over all such random restarts. Construct a PWM from this site-set, and report the PWM and its occurrences sorted by their quality of match (sampling probability).

Update : The goal of the update step is to go from the current site-set C to any new site-set C' such that $\delta(W(C')) > \delta(W(C))$. This is achieved in two steps, as follows. (Also see ''Explanation''.)

(1) **DELETE STEP**: For each site $c \in C$, compute the score $\delta(W(C - \{c\}))$. Choose the c that gives the highest such score, and delete it from C to obtain a site-set C^{del}, of cardinality $n - 1$.

(2) **ADD STEP**: For each l_m-length substring s of each sequence in S^+ (on both strands), let C_s^{add} represent the result of adding s to C^{del}. Estimate the score of C_s^{add} as

$$\delta_{est}(W(C_s^{add})) = \delta(W(C^{del})) + \sum_{k,\alpha} \left(\left. \frac{\partial \delta(w)}{\partial w_{k\alpha}} \right|_{w = W(C^{del})} \times [(W(C_s^{add}))_{k\alpha} - (W(C^{del}))_{k\alpha}] \right) \tag{3}$$

Sort all s in descending order of δ_{est} to obtain a list Λ. Traverse this list, and for each encountered s, compute the exact value of $\delta(W(C_s^{add}))$. If this computed score is better than the score $\delta(W(C))$ before the delete-step, update the current site-set to C_s^{add}, and stop the list traversal. In the

implementation, the list traversal stops after 500 single-site additions have been examined without improvement in score.

Explanation: The update step deletes one site from the current site-set C, and replaces it with a site (substring of some sequence in S^+) that improves the discrimination-score δ. The deletion step follows steepest ascent hill climbing—try all possible single-site deletions, and choose the one that gives the highest δ score, obtaining C^{del}. The addition step is a simple hill climbing heuristic, which attempts to find *any* single-site addition that causes a net increase in the δ score *over that before the deletion step*. However, unlike the deletion step, all possibilities C_s^{add} are not evaluated exactly before deciding, since exact computation is an expensive operation. (See below.) Instead, all possibilities are quickly *estimated* using derivative information—for each possible single-site addition to C^{del}, the change in $W(C^{del})$, the $4l_m$-dimensional vector (PWM) corresponding to C^{del}, is computed, and the score of the new PWM is estimated by using the partial derivative information in each dimension (k, α). (This estimation ignores quadratic and higher order terms in the Taylor series expansion of δ about $W(C^{del})$.) By then sorting the possibilities C_s^{add} on their estimated score, more promising candidate motifs are brought to the front of the list Λ. This drastically reduces the number of expensive exact evaluations done before finding an improvement, and also leads to more optimal moves. Note that if the estimated new scores for each possible addition were accurate, then the estimation step would automatically produce the best (greedy) choice—the head of the sorted list Λ. However, the estimations are not accurate due to the linear approximation made; hence the list Λ may have to be traversed beyond the head, with an exact evaluation of $\delta(W(C_s^{add}))$ for each traversed site s, before finding a score-improving choice.

Let L_{total} be the total length of all input sequences. Each exact evaluation of δ is an $O(L_{total}l_m)$ operation using the Forward-Backward algorithm for w-score calculations. The calculation of δ_{est} is an inexpensive operation, requiring only $O(L_{total}l_m^2)$ time for all possible single-site addition moves. (See Appendix.) By using these δ_{est} values, we are able to induce an order on the exact δ evaluations that reduces the number of evaluations made before a score-improving choice is found. Our experiments with the algorithm (Section 4) revealed that over 90% of the ADD steps require only one exact evaluation. An implementation that did not sort Λ by δ_{est} typically required 22 times as many evaluations per successful ADD step, and took 3 times as many updates (moves) to find a solution whose score was typically about 0.6 times that reported by the default implementation. The reason for splitting the update into two separate steps (delete and add) is that this causes the change in $W(C)$ to be smaller for each possible move, giving better estimates δ_{est}. A special move, described in the Appendix (''big move''), is executed when the ADD step fails to find an improvement.

An A^* algorithm that uses backtracking was implemented and found to give better solutions in some runs. Use of this algorithm is available as an option in the code. We also implemented alternative algorithms such as Gibbs sampling and Conjugate Gradients optimization, which fail due to reasons explained in Section 5.

Initialization of Parameters : The motif cardinality n is set to 20. The background PWM w^b is trained from a user-specified background sequence. (Higher order Markov models may also be specified for the background.) An optional user input e represents the *a priori* expected number of sites of the PWM in all of S^+. (If not specified, we set $e = |S^+|$.) We then set the model parameter $p_m = e/(\sum_{s \in S^+} L_s)$. There are two implemented ways to specify the model parameters p_i for the known PWMs. One option is to set all p_i except p_b to be equal to p_m. The other option is based on maximum likelihood, and is described in the Appendix. The assigned values of the model parameters are kept fixed during the algorithm's execution.

3.4 Derivative computation

Here we provide a rough outline of how derivatives of the w-score are computed. (See Appendix for details.) The w-score defined in Equation 1 is the expectation of the random variable $\chi_m(T)$ over the conditional distribution on parses, $Pr(T|S, \Phi)$. By linearity of expectation, we may express this expectation as $E(\chi_m) = \sum_{i=1}^{|S|} E(\chi_{mi})$, where χ_{mi} is an indicator (0/1) variable for the presence of w^m at position i in a particular parse. Obviously, $E(\chi_{mi}) = Pr(\chi_{mi} = 1)$, and this probability can be computed by using the Forward-Backward algorithm (Durbin *et al*., 1998). A derivative of this probability with respect to any PWM entry $w_{k\alpha}^m$ can also be computed by a similar Forward-Backward algorithm, with the same time complexity, and the derivative of the w-score follows. Thus, computation of $\frac{\partial\sigma(w^m, S, \Phi)}{\partial w_{k\alpha}^m}$ has $O(Ll_m)$ time complexity, implying that all the $4 \times l_m$ partial derivatives of the discrimination score δ may be computed in $O(L_{total}l_m^2)$ time. Once all partial derivatives have been computed, the δ_{est} values from Eqn. 3 can be computed for all possible single-site additions to C^{del} in $O(L_{total}l_m)$ time, by pre-computing the change in δ due to addition of any given base α in any column k of the PWM. (See Appendix.)

4 RESULTS

4.1 Synthetic data

We first performed experiments on randomly generated sequence data, with artificially planted motif instances, to get an insight into the algorithm's idealized performance under controlled conditions. In any experiment, 20 ''positive'' and 20 ''negative'' sequences, of length 400 bp each, were generated randomly. A target motif (PWM) of length $l = 8$ was randomly chosen with a fixed ''relative entropy'' (with respect to background) of R bits per column on average. (R is an experiment parameter. Relative entropy is measure of the column's specificity.) l-mers were sampled from the target PWM according to the sampling probability, and planted at random locations, *only in the positive sequences*, so that the sum of the w-scores of the target PWM was $20n$. (n is an experiment parameter, representing the average w-score per sequence.) Finally, the top $20n$ occurrences of the target PWM, based on their match quality, were noted as target motif occurrences. Each tested algorithm was made to report the top $20n$ occurrences of its optimal motif. The performance score of an algorithm is the number of its reported sites that overlap (at least 6 out of 8 bp) target motif occurrences, as a fraction of $20n$. Since the numbers of target and reported motif occurrences are the same, this represents both the sensitivity and the specificity. (In some experiments with the DME program of Smith *et al*., 2005 below, sensitivity and specificity were different since the program predicted less than $20n$ sites, and a harmonic mean of sensitivity and specificity is reported.)

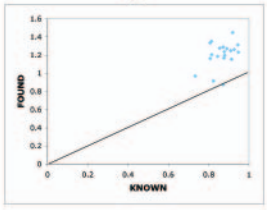
Effect of algorithm settings: In the first set of experiments, we compared two different versions of our algorithm on the same data set. We refer to the algorithm as described in Section 3.3 as **''DIPS''** (**Di**scriminative **P**WM **S**earch). Five random restarts were used by the algorithms in each run. We set $n = 1$ and $R = 1.5$, and performed 20 replicates of each of the following comparisons. These values (of n and R) are typical of PWMs and their w-scores seen in known enhancers involved in the segmentation of the early fruitfly embryo. (Data not shown.)

- DIPS was compared to a conjugate gradients (CG) search algorithm, using the same objective function, and the same number of random restarts. DIPS significantly outperformed CG on 17 replicates; CG was better on one replicate, and both algorithms failed in two cases. Conjugate gradients search was found to get stuck on low-scoring local optima in 15 of 20 replicates.
- Two versions of DIPS, one using the t-score as the discriminative score, and the other using the logistic-score (defined

A

Experiment	e=10	e=15	e=20	e=25	e=30
1	0.75	0.85	0.75	0.75	0.75
2	0.9	0.85	0.85	0.9	0.7
3	0.85	0.85	0.8	0.8	0.85
4	0.85	0.85	0.85	0.75	0.8
5	0.8	0.8	0.8	0.7	0.7
6	0.25	0.3	0.3	0.3	0.3
7	0.75	0.75	0.75	0.75	0.7
8	0.85	0.75	0.75	0.75	0.7
9	0.7	0.6	0.6	0.7	0.6
10	0.7	0.7	0.7	0.75	0.55
Average	**0.74**	**0.73**	**0.72**	**0.72**	**0.67**

B



C

	DIPS	MEME	LP	DME
R=1.25; n=1.0	0.44	0.24	0.36	0.44
R=1.25; n=1.5	0.66	0.47	0.52	0.57
R=1.25; n=2.0	0.79	0.64	0.75	0.71
R=1.5; n=1.0	0.57	0.45	0.62	0.90
R=1.5; n=1.5	0.85	0.55	0.78	0.88
R=1.5; n=2.0	0.9	0.59	0.82	0.92

D

	DIPS	MEME	LP	DME
R=1.25; n=1.0	0.41	0.03	0.34	0.20
R=1.25; n=1.5	0.72	0.09	0.5	0.16
R=1.25; n=2.0	0.81	0.11	0.65	0.28

Fig. 2. A: Performance of algorithm DIPS as a function of the input parameter e, the expected number of sites. Each row represents a data set. B: Comparison of the δ score (re-scaled) of planted (''KNOWN'') and reported (''FOUND'') motifs for each of 20 data sets. The bold line represents FOUND=KNOWN. C: Comparison of performance among DIPS, MEME, LearnPSSM (''LP'') and DME, in the presence of a distractor motif. Each row represents a particular combination of experiment parameters n and R. D: Comparison of performance between DIPS, MEME, LearnPSSM (''LP'') and DME, where the distractor motif is known to DIPS. (Each figure in C & D is an average over 20 experiments. Performance score of DME is the harmonic mean of its average sensitivity and specificity; see text.)

in Appendix), were compared. The t-score implementation outperformed the logistic-score on 9 of the 20 replicates; the logistic-score performed better on 3, and 8 replicates were inconclusive. This is expected since the motif instances were planted in randomly chosen positive sequences, and were not necessarily uniformly distributed among the sequences. Thus, the planting procedure was more akin to the model assumed by the t-score.

- Two versions of DIPS were compared—one in which a candidate motif was allowed to include overlapping substrings from positive sequences (default), and one in which this was disallowed. Both versions performed equally well.

Insensitivity to model parameter p_m: We ran the algorithm DIPS with different input values of the parameter e that represents the expected number of sites, and that is used to determine the parameter p_m. (See ''Initialization of Parameters'', Section 3.3.) While the true value of e was 20, we ran DIPS with $e = 10, 15, 20, 25, 30$, separately on the same data. Ten replicates of the comparison were done, and ten random restarts were allowed in each program run. Figure 2A shows the performance score as a function of e, for each experimental replicate. For most replicates, the performance is comparable across the different values of e. This is an important observation—it shows that the algorithm performance is not very sensitive to the prior expectation of number of sites, i.e., to the model parameter p_m.

Reported score versus planted motif score: The optimal value of the discrimination score, as reported by DIPS, was compared to the discrimination score of the target PWM. In all 20 experimental replicates, the reported score was better than the target PWM's score, typically by a factor of 1.3–1.5. (Fig. 2B.) This shows that the search algorithm actually always finds a motif that is as good or better than the planted motif, in terms of the algorithm's objective function.

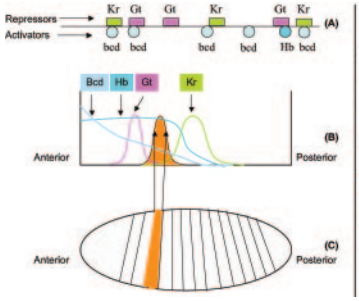
Comparison to alternative motif finding programs: We compared DIPS to a popular non-discriminative motif-finder, MEME (Bailey and Elkan, 1995), that was run on the positive sequences, and made to search for a motif with $20n$ sites. A ''distractor'' motif was also planted, mimicking the planting of the target PWM, except that the distractor was planted in both the positive and negative sets of sequences, in the same amount as the target PWM ($20n$). We expected that MEME, running on the positive sequences alone, will be confounded by the distractor motif, while DIPS will correctly identify the target motif as the true ''discriminative'' motif. We performed 20 experimental replicates for each combination of the experiment parameters $R = 1.25, 1.5$ and $n = 1, 1.5, 2$. The results are shown in Fig. 2C. The average performance of DIPS is significantly higher than that of MEME for all experimental settings.

We also compared DIPS to two discriminative PWM searchers, LearnPSSM (Segal *et al*., 2003) and DME (Smith *et al*., 2005). (The LOGOS program (Xing *et al*., 2003), while similar in the underlying model, is not a discriminative motif finder.) LearnPSSM takes into account both positive and negative sequences, and typically performs comparably to DIPS (and better than MEME), except for the weak-motif experiments ($R = 1.25$, $n = 1, 1.5$) where DIPS did significantly better. This is presumably because with weak motifs there is a greater advantage to a model (DIPS) that rewards multiple occurrences in the same sequence over a model that allows exactly one occurrence (LearnPSSM) per sequence. DME (Smith *et al*., 2005) performs better than all other methods when the motif is highly specific ($R = 1.5$), but its performance takes a hit when the motif is weak ($R = 1.25$, $n = 1.5, 2.0$), where DIPS performs significantly better. We noticed that this drop in performance actually reflected a drop in sensitivity, not specificity, and this was because the motif space searched was limited to PWMs with high information content. (See Appendix for details of how DME was run.)

We performed a second kind of experiment, where a distractor motif was planted only in positive sequences, and was made available to DIPS as a known motif. LearnPSSM was not informed of this distractor motif. (We tried masking out occurrences of the distractor motif before input to LearnPSSM, but the program crashed on such input.) Instead, LearnPSSM was made to report two motifs, and the performance score of the better of the two was taken. We also included the program DME in these comparisons. Since DME cannot take a known motif as prior knowledge, we treated it similarly to LearnPSSM, i.e., it was made to report two motifs, and the better performance score taken. (DIPS was made to report only one motif.) DIPS was found to perform significantly better than both LearnPSSM and DME in these experiments. (Fig. 2D). Due to our limited experience with LearnPSSM and DME, it is possible that the optimal choice of parameters was not made for these programs, and a rigorous comparison is beyond the scope of this paper (Tompa *et al*., 2005). For instance, the performance of LearnPSSM improves when using seed words identified by the SeedSearcher program (Yoseph Barash, personal communication); the same seeds may be used for DIPS also.

4.2 Segmentation network in fruitfly embryo

We next applied our algorithm to cis-regulatory modules (CRM's) involved in segmentation of the early fruitfly embryo. Each CRM is known to drive gene expression in a particular domain



(D)

Phase	Found motif	Best match	Pvalue of match
1 (activator)	anterior.1	**bicoid**	0.00
	posterior.1	**caudal**	0.03
	terminal.1	**torRE**	0.00
	centrall.1	**hunchback**	0.00
2 (repressor)	5.1.1	torRE	0.10
	5.2.1	caudal	0.06
	5.3.1	kruppel	0.25
	5.4.1	**knirps**	0.03
	5.5.1	Dichaete	0.07
3 (activator)	anterior.2	**huckebein**	0.02
	posterior.2	**pdm1_2**	0.00
	terminal.2	caudal	0.06
	centrall.2	huckebein	0.12
4 (repressor)	5.1.2	**knirps**	0.01
	5.2.2	**torRE**	0.01
	5.3.2	giant	0.09
	5.4.2	**giant**	0.05
	5.5.2	bicoid	0.18

Fig. 3. A, B, C: Example of CRM action in fruitfly embryo. A: CRM with binding sites for transcription factors bcd, Hb (activators) and Kr, Gt (repressors). B: Concentration profiles of each transcription factor along the A-P axis. The shaded region is where the CRM drives expression. Bcd and Hb activate, Gt represses from anterior and Kr represses from posterior. C: The domain of expression driven by the CRM, in the embryo. D: Motifs discovered in each phase, and for each expression domain; the best matching known motif, and the p-value of this match.

along the embryo's anterior-posterior axis. This is achieved by the combined action of multiple activators and repressors, whose binding sites are harbored by the CRM. (See Fig. 3A-C.) The goal was to discover PWMs for these binding sites using knowledge of the CRMs' expression patterns. We started with a comprehensive set of 51 CRM's, of median length 1383 bp (Schroeder *et al*., 2004). While many of the involved activators/repressors and their expression domains are well-known, we overlooked this information in this illustrative exercise, to simulate the typical application where the target genes/CRM's and their expression patterns are available and the transcription factors are unknown.

We defined ''activator domains'', i.e., our guesses for domains where the activators are present, as broad regions such as ''anterior'' (50−100% egg length along the anterior-posterior (AP) axis, measured from the tail), ''posterior'' (0−50% egg length), ''terminal'' (0−20% e.l.,80−100% e.l.), and ''central'' (30−70% e.l.). This is in tune with the belief that the early stage activators are maternally deposited proteins with broad expression domains. Repressors in this pathway are believed to have ''gapped'' patterns (one or two bands of expression, about 20% e.l. long, at various positions along the AP axis). Therefore, we defined ''repressor domains'', i.e., our guesses for domains where repressors are present, as successive fifths of the axis: ''5.1'' (80−100% egg length), ''5.2'' (60−80%), ''5.3'' (40−60%), ''5.4'' (20−40%), and ''5.5'' (0–20%). For each defined domain, a ''positive'' set of CRM's and a ''negative'' set was obtained, as follows. For an activator domain, CRM's driving expression in the domain were labeled positive and the rest were labeled negative. For a repressor domain, CRM's driving expression in the domain were labeled negative, and CRM's driving expression in the flanking domains were labeled positive. On average, each data set (of positive and negative sequences) included 22 CRM's. Our algorithm was made to report motifs for each such data set, in phases. In the first phase, each activator domain was analyzed, and in the second phase, each repressor domain was analyzed. The third and fourth phases were repeats of the first two, in that order. The top reported motif from each domain in each phase was input as prior knowledge in the following phases. Thus, a total of $4 + 5 + 4 + 5 = 18$ motifs were obtained, each of length 9. These are tabulated in Figure 3D. Each reported motif was compared to a small compendium of 14 experimentally determined PWMs (Schroeder *et al*., 2004), using the relative entropy (per column) as the similarity score. A p-value was assigned to each similarity score, using 1000 random permutations of the entries of the reported motif. In Fig. 3D, reported motifs that match some known motif with a p-value of 0.05 or less are shown in bold. All such matches to known motifs are consistent with the literature on this pathway (Schroeder *et al*., 2004), as discussed next.

Phase 1,3 (activators): Motif anterior.1 matches the PWM of Bicoid, the known anterior activator. Similarly, posterior.1 and terminal.1 match known motifs of Caudal and torRE, which are known posterior and terminal activators respectively. Motif central.1 matches that of Hunchback, known to have activating role in the central domain. Motifs anterior.2 and posterior.2, discovered for activators in the third phase, match known motifs of Huckebein and Pdm_12, that are known activators in the anterior and posterior domains respectively. Note that different motifs (Huckebein and Pdm_12, versus Bicoid and Hunchback respectively) are discovered in Phases 3 and 1, on the same data sets, showcasing the iterative incorporation of known motifs into the w-score.

Phase 2,4 (repressors): Motifs 5.4.1 and 5.1.2 were discovered when searching for a repressor in domains ''5.1'' (80−100% egg length) and ''5.4'' (20−40%) respectively. They both match the PWM of Knirps, known to be a repressor expressed at 87−100% e.l and at 25−45% e.l., i.e., the same domains where the motif was found. An important repressor, Giant, is the best match for motif 5.4.2, but visual inspection revealed a few differences between the known and reported motifs. Giant is an appropriate transcription factor for the domain ''5.4'' (20−40% e.l.), being expressed at 15−33% e.l. and known to be a repressor. The known PWM of Giant was constructed from only eight binding sites, and is therefore poorly characterized. This may explain the relatively weak resemblance (p-value 0.05) of the Giant PWM to motif 5.4.2. The motif 5.2.2 that matches torRE, and that corresponds to a repressor in domain 60−80% e.l., is presumably an artifact of the torRE (activator) motif present in the terminal (80–100%) CRM's.

Thus, all 10 discovered motifs with a significant match to a known motif, correspond to transcription factors with consistent functionality. These 10 motifs correspond to 8 distinct known motifs, from a compendium of 14 known motifs. Note that this exercise did not utilize the known expression domains of the transcription factors in finding their motifs. We expect that if the positive and negative sequence sets were created based on a factor's precise expression domain, the motif recovery would improve further. The remaining 8 discovered motifs, with insignificant matches to known motifs, are candidates for being novel motifs. In particular, motifs 5.3.1, 5.5.2, corresponding to an unknown repressor in domains ''5.3'' (40−60% e.l.) and ''5.5'' (0−20% e.l.), and the motif central.2, corresponding to an activator in the central domain, are very dissimilar to the known PWMs and deserve further investigation.

4.3 Social behavior in honey bee

Whitfield *et al*., 2003 identified a small set of genes whose expression pattern in whole brain microarray experiments were most discriminative of foraging behavior versus nursing behavior

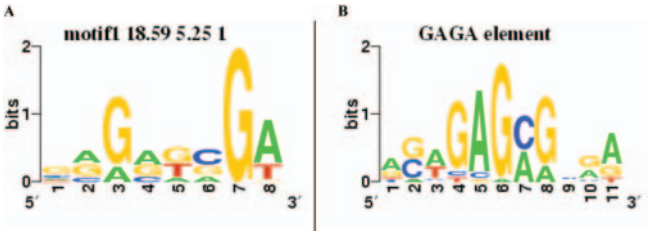


Fig. 4. (A) Motif discovered as discriminating foraging-related versus nursing-related genes in honey bees. (B) the known GAGA motif.

in honey bees. We analyzed the 2Kbp promoters of these genes (21 up-regulated in foragers, 11 up-regulated in nurses). The motif that was most discriminative of these two sets of promoters is shown in Fig 4A. This motif is very similar to the well characterized GAGA element (Fig 4B). The GAGA binding factor is known to regulate gene expression in Drosophila by modulating chromatin structure. Since foraging and nursing behavior in honey bees are controlled by social conditions, our finding represents an important progress in understanding the molecular basis of social behavior.

5 DISCUSSION

The presented algorithm searches the space of site-sets, while the objective function is defined in terms of PWMs. Using PWM w-scores in input sequences is a more sensitive measure of motif occurrence than scores based on the site-set alone (Jensen and Liu, 2004; Lawrence *et al*., 1993), and is crucial for incorporation of negative sequence information in our framework. On the other hand, the search space of site-sets is a restricted subspace of the space of all PWMs (i.e., R^{4l_m}). This imposes a minimum magnitude on any local move made by the algorithm, and is possibly the reason why the algorithm better avoids local optima than a conjugate gradients search in R^{4l_m}. (See Section 4.) A related issue here is the ''motif cardinality'', i.e., the number of substrings forming a site-set. A large value takes us closer to the R^{4l_m} space, while a very small value makes the search space too small and restrictive. We empirically found a motif cardinality of 20 to work well. Finally, we note that there are other ways to restrict the search space than using the space of site-sets; these were not explored.

The algorithm uses a hill climbing heuristic—it moves to any neighbor that gives an improvement, using derivative information to order the evaluation of possible moves. The exact score computation for any new candidate motif is an expensive operation, linear in the total length of sequences. An alternative strategy such as Gibbs sampling would require evaluating a large number of neighbors (linear in the average length of a sequence) before deciding the move, and is hence impractical in this setting. Another algorithmic choice was to run our hill climbing algorithm followed by conjugate gradients search in the R^{4l} space—we tested this option and found no improvement in performance. The presented algorithm is fairly robust to the choice of the initial random seed. In the experiments of Section 4, the optimal motif was typically obtained in at least two of the five random restarts.

The algorithm is implemented to optimize any differentiable function of the w-scores in individual sequences, and may be trivially modified to use only the positive set of sequences (as in traditional motif finding), or to optimize correlations between sequence and gene expression data. Such modifications and their performance are, however, outside the scope of this paper. The w-score may also be computed in the presence of multiple-species data, using a probabilistic model of binding site evolution (Sinha *et al*., 2004), therefore enabling a phylogenetic version of the algorithm.

The model parameters Φ (including the unknown motif's p_m) define the ''global distribution'' (Xing *et al*., 2003) $Pr(T)$ on motif occurrences. We then use Bayes rule to compute the conditional distribution $Pr(T \mid S)$ on parses, and hence compute an average count of the motif(s) *given* the sequence data. One option that was not explored is to include p_m as a trainable parameter in the search algorithm, i.e., to find the w^m *and* the p_m that maximize the discrimination score. However, this will mean that the count (w-score) of w^m in a sequence S will depend on sequences other than S.

6 FUTURE WORK AND CONCLUSIONS

The choice of the model parameter p_m is *ad hoc*. Further exploration of this issue, such as specification of a probability distribution over p_m, and integration over that distribution, will be important. Similarly, the choice of the optimal motif (site-set) cardinality is a future research direction. We will also explore a probabilistic version of the discrimination score, such as that of Segal *et al*., 2003, while using the w-score to count PWM occurrences.

We have proposed a statistically sound method to ''count'' PWM (motif) occurrences in a given DNA sequence. This count is efficiently differentiable with respect to the PWM parameters, enabling search algorithms to use derivative information for a large class of objective functions. We propose a derivative-guided hill climbing algorithm to find a motif that best discriminates two different sets of sequences by its counts in those sequences. The algorithm is tested on synthetic data and is applied to the segmentation pathway in the fruitfly and on behavioral genes in honey bee to elicit several interesting motifs.

ACKNOWLEDGEMENTS

The author is grateful to Eric Siggia and Eran Segal for useful discussions, to Yoseph Barash for making the LearnPSSM code available for experimentation, and to Andrew Smith for sharing the DME code.

REFERENCES

Bailey,T.L. and Elkan,C. (1995) Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21(1–2): 51–80.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis*. Cambridge University Press.

Hong,P., Liu,S., Zhou,Q., Lu,X., Liu,J.S. and Wong,W.H. (2005) A boosting approach for motif modeling using ChIP-chip data. *Bioinformatics*, 21(11).

Jensen,S.T. and Liu,J.S. (2004) BioOptimizer: a Bayesian scoring function approach to motif discovery. *Bioinformatics*, 20(10).

Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwald,A.F. and Wootton,J.C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262, 208–214.

Schroeder,M.D., Pearce,M., Fak,J., Fan,H., Unnerstall,U., Emberly,E., Rajewsky,N., Siggia,E.D. and Gaul,U. (2004) Transcriptional control in the segmentation gene network of Drosophila. *PLoS Biol*, 2(9).

Segal,E., Yelensky,R. and Koller,D. (2003) Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, 19(S1).

Sinha,S. (2003) Discriminative Motifs. *J. Comput. Bio.*, 10(3−4).

Sinha,S., Blanchette,M. and Tompa,M. (2004) PhyME: a probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics*, **5**(170).

Sinha,S., van Nimwegen,E. and Siggia,E.D. (2003) A probabilistic method to detect regulatory modules. *Bioinformatics*, 19(S1).

Smith,A.D., Sumazin,P., Das,D. and Zhang,M.Q. (2005) Mining ChIP-chip data for transcription factor and cofactor binding sites. *Bioinformatics*, 21(S1): i403–412.

Stormo,G.D. and Fields,D.S. (1998) Specificity, Free Energy and Information Content in Protein-DNA Interactions. *Trends in Biochemical Sciences*, **23**, 109–113.

Sumazin,P., Chen,G., Hata,N., Smith,A.D., Zhang,T. and Zhang,M.Q. (2005) DWE: discriminating word enumerator. *Bioinformatics*, **21**(1).

Takusagawa,K. and Gifford,D. (2004) Negative information for motif discovery. In *Pacific Symposium on Biocomputing*, Hawaii, pp. 360–371.

Tomancakal,P. *et al*. (2002) Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biol*., **3**(12).

Tompa,M. *et al*. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, **23**(1).

Whitfield,C.W., Cziko,A.M. and Robinson,G.E. (2003) Gene expression profiles in the brain predict behavior in individual honey bees. *Science*, **302**(5643), 296–9.

Xing,E.P., Wu,W., Jordan,M.I. and Karp,R.M. (2003) LOGOS: a modular Bayesian model for de novo motif detection. In *IEEE Computer Society Bioinformatics Conference*.

7 APPENDIX

7.1 Sampling probability for PWM

Let $w_{k\alpha}$ denote the probability of base α in column k of a PWM w of length l. The weight matrix induces a probability distribution on all strings of length l. The probability of sampling a string s of length l from the PWM w is defined as $Pr(s \mid w) = \prod_{k=1}^{l} w_{ks_k}$, where s_k is the k^{th} base of s.

7.2 Logistic-score

The ''logistic-score'' is an alternative discrimination score implemented, defined as $\lambda(S^+, S^-, w^m) =$

$$-\left(\sum_{s\in S^+}(1 - logit(\sigma_s/L_s))^2 + \sum_{s\in S^-}(logit(\sigma_s/L_s))^2\right) \qquad (4)$$

where the function $logit(x) = 2/(1+e^{-\omega x}) - 1$ is a rescaled logistic function with range 0 (at $x=0$) to 1 (at $x=\infty$). (L_s for a string s represents its length.)

The logistic-score represents the least-squares error of a classifier that uses the *logit* function as a soft predictor for class membership, with 0 representing S^- and 1 representing S^+. (The negative sign is to minimize the error while maximizing the function.) In the algorithm, during the initialization of parameters, the exponent factor ω in the logistic function is set to $2e/\mid S^+ \mid$. (See ''Initialization of Parameters'', Section 3.3, for definition of e.)

7.3 The ''big move''

The update (Section 3.3) uses two separate hill climbing steps. As such, it may fail to find an improvement even if there exists a deletion choice that is itself non-optimal, but leads to a score improvement in combination with the appropriate addition choice. The ''big move'' is executed when the above default procedure fails. This move effects each possible deletion, computes derivatives, and estimates the score of each possible addition. It then sorts all such deletion-addition pairs by estimated score, and serially evaluates each pair, stopping when an improvement is found, or when a certain number of evaluations (500, for the current implementation) have been made.

7.4 Incorporating known PWMs into the score

The algorithm assigns the transition probabilities for the known motifs $W_{known} = \{w^1, \ldots w^\kappa\}$ separately for each sequence S, as follows. It first assumes that the only PWMs in the model are w^b and the set W_{known}, and computes the values of the corresponding p_i parameters that maximize the likelihood of the sequence S. This computation, described in Sinha *et al*. 2003, uses an Expectation-Maximization algorithm and is known to have good convergence property. The transition probabilities p_i, for $i = 1 \ldots \kappa$, are then fixed at these trained values. The transition probability p_m for the desired motif w^m is then assigned as described in Section 3.3, and p_b is obtained from the constraint $(\sum_{j=1}^{\kappa} p_j) + p_m + p_b = 1$. Note that this step causes little change in p_b, since p_m is small; hence the values of $p_1, p_2, \ldots p_\kappa$ and p_b are those determined by the maximum likelihood inference.

7.5 Computing derivatives of w-score

The w-score of PWM w^m in sequence S of length L with model parameter p_m is defined in Equation 1. We re-write the definition with a slight change of notation as

$$\sigma_m(S) = \sum_T \chi_m(T) Pr(T \mid S, \Theta)$$

where $\chi_m(T)$ is the number of times w^m is planted in parse T, and Θ represents the model parameters. In the general case, this includes w^m and its transition probability p_m, the background motif w^b (with p_b), and the set of known motifs $W_{known} = \{w^1, w^2, \ldots w^\kappa\}$ with corresponding transition probabilities $p_1, p_2, \ldots p_\kappa$, with the constraint $(\sum_{j=1}^{\kappa} p_j) + p_b + p_m = 1$. Let the indicator (0/1) variable $X_{ml}(T)$ be 1 if motif w^m is planted at position l in parse T. Then $\sigma_m(S)=$

$$\sum_{l=1}^{L}\sum_{T\mid X_{ml}(T)=1} Pr(T \mid S, \Theta) = \sum_{l}\sum_{T\mid X_{ml}(T)=1} \frac{Pr(S,T \mid \Theta)}{Pr(S \mid \Theta)} \qquad (5)$$

Therefore, we can write the partial derivative of $\sigma_m(S)$ as $\frac{\partial \sigma_m(S)}{\partial w^m_{k\gamma}} =$

$$\frac{\frac{\partial}{\partial w^m_{k\gamma}}\sum_l \sum_{T\mid X_{ml}(T)=1} Pr(S,T\mid\Theta)}{Pr(S\mid\Theta)} - \frac{\sigma_m(S)\frac{\partial}{\partial w^m_{k\gamma}} Pr(S\mid\Theta)}{Pr(S\mid\Theta)} \qquad (6)$$

Let B and C denote the two terms in the sum on the right hand side. Let us define the ''forward'' variable $\alpha(l)$ as the probability of generating the subsequence $S[1\ldots l]$ by the model, such that some w^i ends at l. Similarly, let the ''backward'' variable $\beta(l)$ be the probability of generating the subsequence $S[l\ldots L]$ by the model, such that some w^i begins at l. Let $\alpha'(l) = \frac{\partial \alpha(l)}{\partial w^m_{k\gamma}}$ and $\beta'(l) = \frac{\partial \beta(l)}{\partial w^m_{k\gamma}}$. By definition, $\alpha(L) = Pr(S \mid \Theta)$, hence we have $\frac{\partial}{\partial w^m_{k\gamma}} Pr(S \mid \Theta) = \alpha'(L)$. This gives us

$$C = \frac{\sigma_m(S)\alpha'(L)}{\alpha(L)} \qquad (7)$$

$$B = \frac{1}{\alpha(L)} \sum_l \frac{\partial}{\partial w^m_{k\gamma}} \sum_{T\mid X_{ml}(T)=1} Pr(S,T \mid \Theta) \qquad (8)$$

Now, the term $\sum_{T\mid X_{ml}(T)=1} Pr(S,T \mid \Theta)$ may be expressed using the forward and backward variables as being equal to $\alpha(l-1)p_m Pr(S[l\ldots l+l_m-1] \mid w^m)\beta(l+l_m)$, where l_m is the length of w^m. Hence we have

$$\begin{aligned}
&\tfrac{\partial}{\partial w^m_{k\gamma}} \textstyle\sum_{T\mid X_{ml}(T)=1} Pr(S,T\mid\Theta) \\
&= \alpha'(l\text{-}1)p_m Pr(S[l\ldots l+l_m\text{-}1]\mid w^m)\beta(l+l_m) \\
&+ \alpha(l\text{-}1)p_m[\tfrac{\partial}{\partial w^m_{k\gamma}} Pr(S[l\ldots l+l_m-1]\mid w^m)]\beta(l+l_m) \\
&+ \alpha(l-1)p_m Pr(S[l\ldots l+l_m-1]\mid w^m)\beta'(l+l_m)
\end{aligned} \qquad (9)$$

where $\frac{\partial}{\partial w^m_{k\gamma}} Pr(S[l\ldots l+l_m-1]\mid w^m) = \prod_{j=1}^{l_m} (w^m_{j\gamma})^{(1-\delta_{jk})}$, and δ_{jk} is the Kronecker delta function. Finally, we consider the derivatives

of the forward and backward variables. Since $\alpha(l) = \sum_i \alpha(l - l_i)p_i Pr(S[l - l_i + 1 \ldots l] \mid w^i)$, where l_i is the length of w^i, we can write $\alpha'(l) =$

$$[\sum_i \alpha'(l - l_i)p_i Pr(S[l\text{-}l_i + 1 \ldots l] \mid w^i)]$$

$$+ \alpha(l - l_m)p_m \frac{\partial Pr(S[l - l_m + 1 \ldots l] \mid w^m)}{\partial w^m_{k\gamma}} \qquad (10)$$

and $\beta'(l)$ is obtained by a similar recursion. Combining Equations 7, 8, 9, 10, and replacing into Equation 6, we obtain the partial derivative of $\sigma_m(S)$ with respect to $w^m_{k\gamma}$.

In the implementation, underflows are handled by using scaling constants, an issue not considered in the above description.

7.6 Time complexity

α' and β' can be computed using a forward and backward algorithm in time $O(L\sum_i l_i)$. Computation of B has the same time complexity, and hence computing each $\partial\sigma_m(S)/(\partial w^m_{k\gamma})$ takes $O(L\sum_i l_i)$ time, giving an $O(Ll_m\sum_i l_i)$ time complexity for the entire derivative computation of the w-score. Since this computation has to be done for each sequence in $S^+ \cup S^-$, the total time complexity is $O(L_{total}l_m\sum_i l_i)$. If the only PWMs are w^m and w^b, this reduces to $O(L_{total}l_m^2)$.

Knowing all $4l_m$ partial derivatives, we can compute the δ_{est} for all single-site additions s to the current site-set C^{del} as follows. Note that any site addition $s = s_1 s_2 \ldots s_{l_m}$ changes the PWM $W(C^{del})$ in a particular way: in column k, the frequency (integral count) of base s_k goes up by 1, and the frequency of the other bases in that column remains same. Thus, there are only four possibilities for the (vector) change in the k^{th} column, regardless of s. Hence there are only four possibilities for the term $\sum_\alpha (\frac{\partial\delta(w)}{\partial w_{k\alpha}} \mid_{w=W(C^{del})} \times [(W(C_s^{add}))_{k\alpha} - (W(C^{del}))_{k\alpha}])$, regardless of s. We can pre-compute each of these four possibilities, for each k. Then, for any single-site addition s, we only have to look up l_m of these pre-computed values (one for each k) and sum them to obtain $\delta_{est}(W(C_s^{add}))\text{-}\delta(W(C^{del}))$, in $O(l_m)$ time. Thus, computing the δ_{est} for all possible single-site additions takes $O(L^+l_m)$ time, where L^+ is the total length of all positive sequences, and hence the maximum number of single-site additions s.

7.7 Experiments on synthetic data

Our program (DIPS) was run with the '-len' option set to the desired motif length, with ''-niter 5'' to try 5 random initial seeds per motif, and ''-nsites'' set to the $20n$, which is the number of sites planted.

LearnPSSM was run with seed length (-l) and PSSM length (-L) both set to the desired motif length, with the '-r' option to search both strands, and with ''-m 1000'' to try 1000 seeds for each motif. The ''Training'' file assigned a weight of 0.99 to each sequence in the positive set, and a weight of 0.01 to each sequence in the negative set.

DME was run with motif width (-w) set to the desired motif length, and the ''minimum number of bits per column'' (-i) set to 1.5. We also experimented with setting the '-i' option equal to the bits per column of the true (planted) motif, and found the results to be poorer for weak motifs (-i 1.25), and hence report the better results (from using -i 1.5).

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e464–e471
doi:10.1093/bioinformatics/btl207

An ontology for a Robot Scientist

Larisa N. Soldatova*, Amanda Clare, Andrew Sparkes and Ross D. King

Department of Computer Science, The University of Wales, Aberystwyth, Penglais, Aberystwyth, SY23 3DB, Ceredigion, UK

ABSTRACT

Motivation: A Robot Scientist is a physically implemented robotic system that can automatically carry out cycles of scientific experimentation. We are commissioning a new Robot Scientist designed to investigate gene function in *S. cerevisiae*. This Robot Scientist will be capable of initiating >1,000 experiments, and making >200,000 observations a day. Robot Scientists provide a unique test bed for the development of methodologies for the curation and annotation of scientific experiments: because the experiments are conceived and executed automatically by computer, it is possible to completely capture and digitally curate all aspects of the scientific process. This new ability brings with it significant technical challenges. To meet these we apply an ontology driven approach to the representation of all the Robot Scientist's data and metadata.

Results: We demonstrate the utility of developing an ontology for our new Robot Scientist. This ontology is based on a general ontology of experiments. The ontology aids the curation and annotating of the experimental data and metadata, and the equipment metadata, and supports the design of database systems to hold the data and metadata.

Availability: EXPO in XML and OWL formats is at: http://sourceforge.net/projects/expo/. All materials about the Robot Scientist project are available at: http://www.aber.ac.uk/compsci/Research/bio/robotsci/.

Contact: lss@aber.ac.uk

1 INTRODUCTION

1.1 Our new Robot Scientist

A Robot Scientist is a physically implemented robotic system that applies techniques from artificial intelligence to carry out cycles of scientific experimentation (King *et al*., 2004). A Robot Scientist automatically: originates hypotheses to explain observations; devises experiments to test these hypotheses; physically runs the experiments using laboratory robotics; interprets the results; and then repeats the cycle.

The first Robot Scientist was built in Aberystwyth to investigate *S. cerevisiae* gene function using deletion mutants and auxotrophic growth experiments. In our original proof-of-principle work we demonstrated that a Robot Scientist could rediscover biological knowledge concerning gene function in the aromatic amino acid synthesis pathway. Recently, we have demonstrated that the same approach can be extended to the discovery of novel biological knowledge (King *et al*., 2005).

*To whom correspondence should be addressed.

An important limitation of our Robot Scientist research has been that although all the intellectual steps were automatic, for some experimental steps it was necessary to intervene manually, owing limitations in our robotic equipment. To eliminate this manual intervention we are commissioning a fully automated Robot Scientist (Figures 1 and 2). This new system is designed to automatically execute yeast growth experiments by: selecting frozen yeast strains from a freezer; inoculating these strains into rich medium; then harvesting a defined quantity of cells; inoculating these cells into specified media (base plus added metabolites and/or inhibitors); and finally accurately measuring growth curves by measuring optical density (OD) (King *et al*., 2005). We believe, after consulting with the laboratory automation industry, that our new Robot Scientist is one of the most complicated laboratory automated systems in any academic laboratory.

In constructing this new Robot Scientist we have taken advantage of the key benefit of automation: its ability to be easily scaled up. The new Robot Scientist is designed to initiate >1,000 new strain/defined growth-medium experiments a day, using a minimum of 50 different yeast strains, with up to 7 metabolites per experiment, and with each experiment lasting up to 3 days (plus an initiation day). Accurate growth curves will be obtained by observing optical density for every experiment every 20 minutes. This will result in >200,000 data measurements a day. In addition, we expect >1,000,000 meta-data measurements each day. These include hypotheses, experimental plans, experimental actions, temperature, humidity, etc.

1.2 Ontologies for curation and annotation of scientific experiments

Robot Scientists provide unsurpassed test beds for the development of methodologies for the curation and annotation of scientific experiments. This is because, as the experiments are conceived and executed automatically by computer, it is possible to completely capture and digitally curate all aspects of the scientific process: the hypotheses, the experimental goals, the results, etc. The use of a Robot Scientist removes the often 'show stopping' sociological problems associated with trying to capture such data from human scientists.

The ability to capture all relevant experimental information brings with it significant technical challenges:

- We require a very detailed and formalised description of all the domains involved in an experiment: experimental design, methods and technologies; experimental object models and

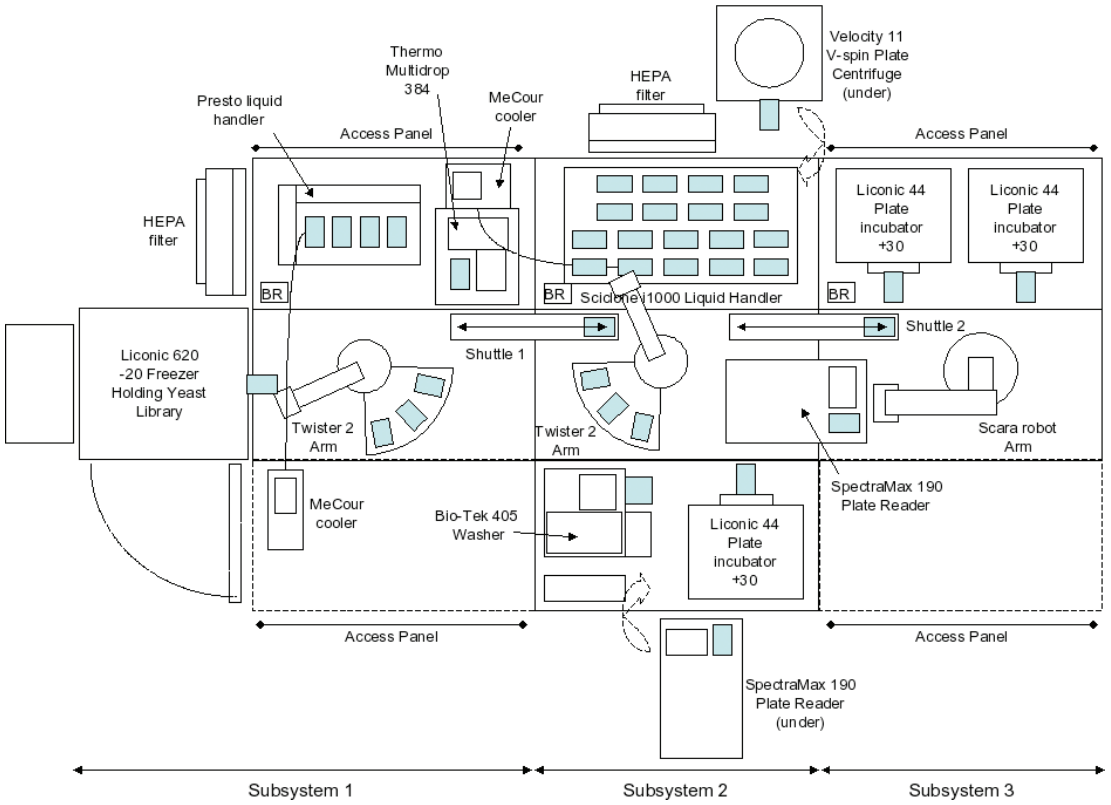


Fig. 1. Plan of our new Robot Scientist.

Fig. 2. Our new Robot Scientist (during assembly, Nov., 2005).

background knowledge; reasoning rules for analysis of the experimental results, etc.

- We need to curate and ensure the integrity of the large amount of data and metadata that the Robot Scientist will produce.
- We wish to make the experimental information as open as possible to both the scientific community and the general public—as part of the mission to improve the public understanding of science.

To meet these challenges we have selected an ontology driven approach to the representation of all the data and metadata relevant to the project. The value of the utilisation of ontologies for the curation and annotation of scientific results is now generally recognised (Bard and Rhee, 2004). The use of ontologies make scientific knowledge more explicit, helps detect errors, enables the sharing and reuse of common knowledge, removes redundancies in domain-specific ontologies, and promotes the interchange and reliability of experimental methods and conclusions.

Bioinformatics has led the way in the application of ontologies to the curation and annotation of experimental data (Brazma *et al*., 2001). Probably the best known application of ontologies to describing experiments is that developed by the Microarray Gene Expression Society (MGED) (Stoeckert *et al*., 2002). The MGED Ontology (MO) is designed to provide descriptors required by MIAME (Minimum Information About a Microarray Experiment) standard for capturing core information about microarray experiments. MO aims to provide a conceptual structure for microarray experiment descriptions and annotation. Similar approaches have been made in proteomics (http://psidev.sourceforge.net/ontology/), metabolomics (Jenkins *et al*., 2004) and anatomy (Ryn and Sternberg, 2003).

Unfortunately, the existing ontologies for experiments representation are not suitable for extension to a Robot Scientist (Soldatova and King, 2005). They are highly human-oriented, and they do not contain concepts about general principles for organising and execution of experiments and analysis of the results. In

addition, no ontology is yet available for microbiological experiments, the domain of the robot scientist experiments.

We have therefore applied our generic ontology of scientific experiments EXPO (Soldatova, 2005) to our Robot Scientist and formed the instantiation EXPO-RS. The goals of this ontology are as follows:

- To formalise the concepts involved in Robot Scientist experiments, and to identify what metadata are essential for the experiment's description and repeatability.
- To provide a controlled vocabulary for all the participants of the project. This includes specialists from different scientific areas (and the general public).
- To organise all the information and knowledge about the Robot Scientist project into different meta-levels. This ensures a clear structure, allows maintenance and updating of the knowledge, and enables coordination of multiple tasks: planning of an experiment; execution of an experiment; access to the results; technical support of the robot, etc.
- To design a database for the storage of experimental data and track experiment execution.

In section 2 we describe a generic ontology of experiments as a method for representation of the information about the Robot Scientist project. Section 3 presents three example applications of the ontology for the Robot Scientist description, namely: its metadata, representation of the data about the experimental equipment and the data base model for storing information about Robot Scientist experiments. Section 4 is devoted to discussion of problems of the data representation for a robot and new challenges.

2 GENERIC ONTOLOGICAL DESCRIPTION OF EXPERIMENTS

We used the generic ontology of scientific experiments EXPO as a method to represent the metadata and data of the Robot Scientist experiments (Soldatova and King, 2006). EXPO provides a clear structured framework for a consistent and shareable description of experiments for both humans and computer systems. EXPO formalises the generic concepts of experimental design, methodology, experimental objects, subjects, equipment, experimental protocols and actions, observations and results representation. EXPO is expressed in the W3C standard ontology language OWL-DL (www.w3.org/TR/owl-guide/). EXPO contains 200 classes and it is available at http://sourceforge.net/projects/expo/.

In defining an ontology, we follow the definition given by Barry Smith[1]: An ontology is a representation of some pre-existing domain of reality which: (1) reflects the properties of the objects within its domain in such a way that there obtains a systematic correlation between reality and the representation itself; (2) is intelligible to a domain expert; (3) is formalised in a way that allows it to support automatic information processing.

To build up the Robot Scientist's ontology EXPO-RS we use the following structure elements:

- **A concept X (=class)**. 'X is a class if and only if (iff) each element x of X satisfies the intrinsic property of X. The intrinsic property of a thing is a property which is essential to the thing and it looses its identity when the property changes' (Mizoguchi, 2004).
- **An instance x**, an element of the class X.
- **Is-a relation**. '<class A is-a class B> relation holds between classes if and only if (iff) every instance of the class A is also an instance of the class B' (Mizoguchi, 2004). In order to provide a simple hierarchical structure, the concepts are assumed to be disjoint.
- **Instance-of relation**. If and only if (iff) the definition above holds then the relation <x instance-of X> is true.
- **Attribute-of (a/o) relation** is used for describing properties of the concept. It can be considered as a predicate `attribute (Concept, Property)`. This relation can have a fixed cardinality or a range $0, \ldots, n$, where n is a natural number; minimum cardinality 0 means that some of instances of the class might not have this property, i.e. the property is not intrinsic, but still important for the class description as a whole.
- **Part-of relation (p/o)** is used for describing partronomic relations between concepts. For simplicity's sake and because it is not essential for the selected domain, we do not distinguish the different types of whole-part relations (Guarino, 1998). The above comments about cardinality are also true for part-of relations.

All concepts of the Robot Scientist project are defined as subclasses of the following top concepts:

(1) Physical **object**, i.e. experimental equipment.

(2) **Process**, such as an execution of experiment, interpreting the results, experimental actions.

(3) **Proposition**: tasks of experiments, experimental goals, hypotheses, experimental design strategy, models, standards.

(4) **Substrate** for representing time points and intervals, measurement units and locations.

(5) **Role**, for instance functional role, or subject, object role.

The role concept is particularly important for the Robot Scientist because the robot can play different roles in the same experiment:

- The robot is the *object* of an experiment when we study the automation of science. The experimental domain in this case is Artificial Intelligence and Robotics.
- The robot is the *subject* of the experiment when we employ the robot to discover new knowledge in a scientific domain. In this article we concentrate on the description of robot-subject experiments.

EXPO-RS is built as an extension of EXPO by adding the specifics of the Robot Scientist project to the classes and instances.

3 APPLICATIONS OF AN ONTOLOGY FOR THE ROBOT SCIENTIST

3.1 Metadata

We illustrate in Figures 3 and 4 an example of a Robot Scientist experiment annotated using EXPO-RS (King *et al*., 2005). In

[1]The Buffalo Ontology Site: http://ontology.buffalo.edu/

```
<scientific experiment>:
    <admin. info about experiment>:
    <title>:                          Robot scientist
    <ID>:                             exp200401113-0001
<classification by domain>:
    <domain of experiment>:
    <DDC(Dewey) classification>:      576 Microbiology
<research hypothesis>:
<representation style>:               <text>
    <linguistic expression>:          <natural language>:
                                      Knocked out gene named ``yer152c'' (= met8) has the function named
                                      ``2-aminoadipate:2-oxoglutarate aminotransferase'' (E.C.2.6.1.39)
    <linguistic expression>:          <artificial language>:
                                      encodes(yer152c, '2-aminoadipate:2-oxoglutarate aminotransferase')
<null hypothesis>:
    <linguistic expression>:          <artificial language>:
                                      ~ encodes(yer152c, '2-aminoadipate:2-oxoglutarate aminotransferase')
<alternative hypothesis>:
    <linguistic expression>:          <natural language>:
                                      <time effect>: maturation effect (incubator too cold)
<alternative hypothesis>:
    <linguistic expression>:          <natural language>:
                                      <object effect>: no entry of metabolite into the cells
<alternative hypothesis>:
    <linguistic expression>:          <natural language>:
                                      <object effect>: cross contamination
<domain model>:                       <representation style>: <text>
    <linguistic expression>:          <artificial language>: Prolog
                                      A logical model of yeast metabolism
    <reference>:                      Whelan, K.E. & King, R.D. (2005) Using a logical model to predict
                                      the growth behaviour of yeast cell cultures. Department of Computer
                                      Science Report, University of Wales, Aberystwyth. UWA-DCS-05-045.
<experimental design>:
    <subject>:                        The Robot Scientist
    <object>:                         S. cerevisiae
<experimental model>:
    <factor>:                         Strain - 2 strains: wild [Mat A, by4741] and its yer152c knockout
    <factor>:                         addition or not of metabolite 2-aminoadipate:2-oxoglutarate
                                      aminotransferase
    <model assumption>:               stationarity

.............................................
<experimental conclusion>:            <representation style>: <text>
    <linguistic expression>:          <natural language>:
                                      The yer152c knockout strain has a quite different growth profile to
                                      the wild type. This is consistent with yer152c encoding a
                                      2-aminoadipate:2-oxoglutarate aminotransferase. We hypothesize that
                                      yer152c is the missing 2-aminoadipate:2-oxoglutarate
                                      aminotransferase II.
```

Fig. 3. EXPO-RS formalisation of a Robot Scientist experiment in a text format (a fragment).

Figure 3 (and further in the text) the terms in angled brackets are from EXPO-RS. Figure 3 shows the corresponding fragment of EXPO-RS in a text format and Figure 4 in a graphic format (Kozaki *et al*., 2002).

The goal of the illustrated experiment is to investigate the function of the gene named 'YER152c'. This gene is currently classified by SGD/GO as 'Uncharacterized', and by MIPS as 'Unclassified'. In previous work on predicting gene function we predicted the gene to be involved in 'metabolism' with estimated >80% accuracy (Clare and King, 2003).

The Robot Scientist used its background bioinformatics knowledge in its internal databases to abduce the hypothesis that YER152c encodes the enzyme 2-aminoadipate: 2-oxoglutarate aminotransferase. This is formally encoded in the Prolog fact 'encodes (yer152c, '2-aminoadipate: 2-oxoglutarate aminotransferase')'. Given this abduction, and its general model of yeast metabolism, the Robot Scientist deduced that the removal of this gene would produce a strain with reduced growth (a bradytrophic mutant) or no growth (an auxotrophic mutant); and that addition of the metabolite L-2-aminoadipate to the standard defined growth medium would restore growth. Analysis of the experimental results provided evidence that was consistent with YER152c encoding the missing 2-aminoadipate: 2-oxoglutarate aminotransferase II (N.B. it is a known iso-enzyme: (Masuda and Ogur, 1969)).

The application of an ontology to this experiment demonstrates its value in providing the structure for annotating and curating our Robot Scientist's experimental information. Note in particular, the use of the ontology made explicit: the analysis of alternative hypotheses, assumptions about the domain model and possible factors that could affect the experimental results. Finally, as EXPO is a general ontology of scientific experiments, its application provides the framework to link the Robot Scientist's data and metadata to other scientific data and metadata.

3.2 Description of experimental equipment

Our new Robot Scientist's laboratory automation hardware is extremely complicated and comes supplied with substantial amounts of technical description. Application of an ontology helps to define which of the equipment characteristics are most important to describe to ensure experimental reproducibility.

Fig. 4. EXPO-RS representation of a robot scientist experiment (a fragment).

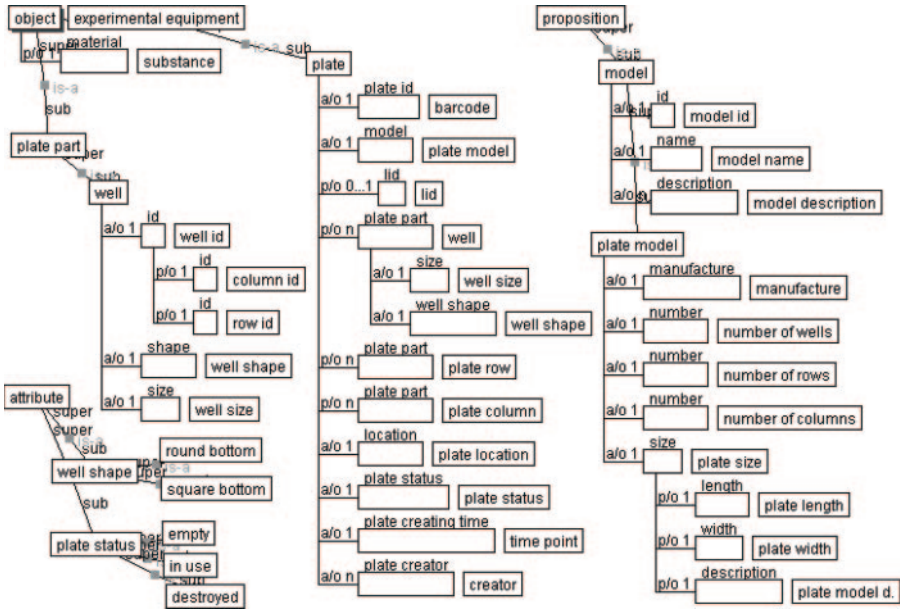


Fig. 5. EXPO-RS representation of the experimental equipment (a fragment for a plate).

A description of the functionality of the equipment highlights the requirement for collection of metadata from the equipment. For example: if the equipment can do an action *A*, do we need to make sure whether or not *A* happened is recorded in our data records; if part of the equipment is replaced due to failure, does the new equipment satisfy the functionality that the old equipment provided, and what are the differences? An ontological description of this functionality gives us a systematic framework for making decisions about the metadata we need to record, and a framework for comparing metadata collected from differing pieces of equipment.

In EXPO-RS each piece of laboratory equipment is defined through 'physical' object. For example, a well is defined as <plate part> (see Figure 5). As a well cannot exist separately

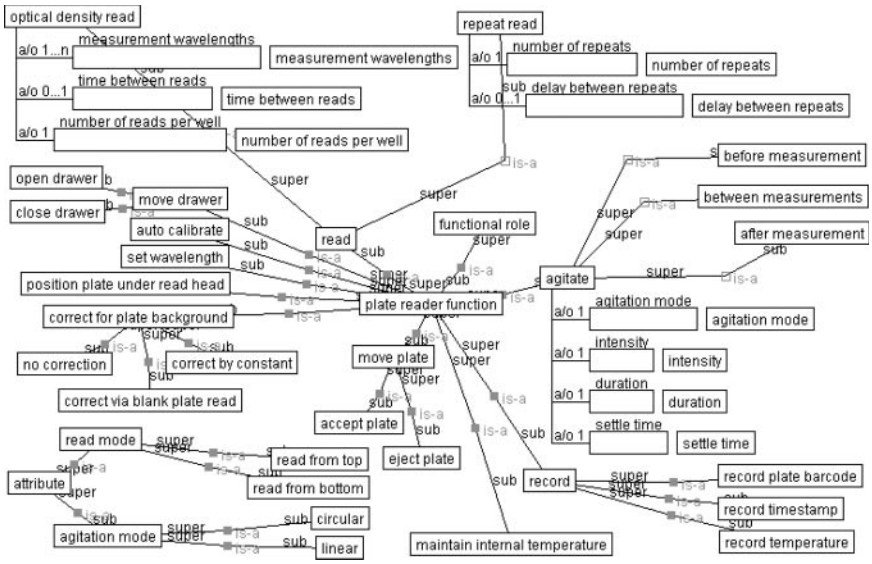


Fig. 6. EXPO-RS representation of a plate reader functions.

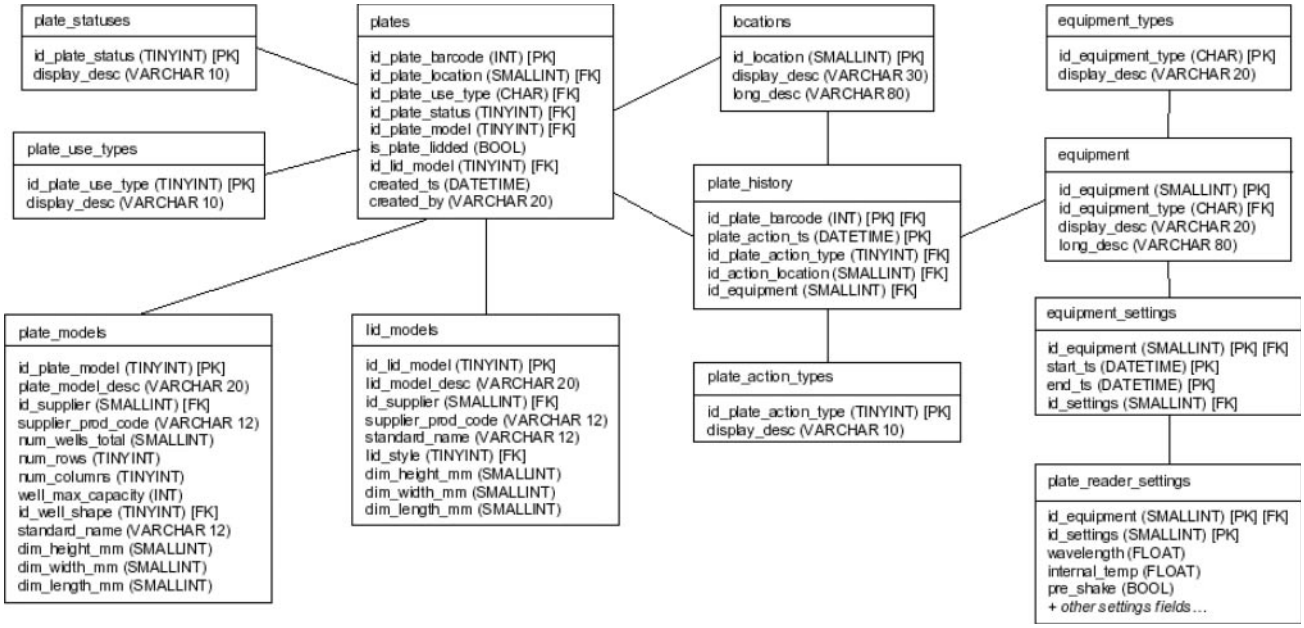


Fig. 7. Data base model for the Robot Scientist (a fragment), where PK is a primary key and FK is a foreign key.

from a plate it cannot be a single object. A representation of a well is essential for representing the Robot Scientist's experimental observations, because optical density is measured in each individual well, and stored by well. The ontology describes a concept <well> with its important characteristics: identification number <well id> (from plate <column id> and plate <row id>); <well shape> that can be <round bottom> or <square bottom>, and <well size>. Note that the attributes <well shape> and <well size> are also used for plate descriptions. The reason for this is that no plate can have wells of differing size and shape. Plates for the pregrowth stage of the experiment will have <round bottom> <well shape> for better centrifugation separation, while those used in the freezer and in the growth phase will have <square bottom> wells.

Administrative information about the equipment, contact details of suppliers and models information are represented as propositions. Each <model> is characterised by its <id>, <name> and has <model description>. The latter can have different <representations> (not shown) on different representation media such as electronic e.g. a CD or paper e.g. a book. <Plate model> inherits properties of <model> concept and additionally has <plate size>, <number of wells> properties, etc. These attributes are not essential in describing a plate as a piece of experimental equipment or for experiment

representation; thus they characterise a particular plate model and are stored separately. A record of the <plate model> and <manufacturer>, and the same for the plate lid (not shown in this fragment) ensures that experimental variation due to readings of different types of plate may be noticed.

The ontology also contains a description of the equipment functionality. We illustrate the application of this ontology by showing a fragment of the functionality of a plate reader. The fragment in Figure 6 shows the main functions of a Robot Scientist plate reader, that is to perform an <optical density read>, to <agitate> the plate, to <record> information such as the plate barcode, timestamp, and temperature, and to allow the correct parameters of the read and agitation to be set. The plate readers currently used in the Robot Scientist are two SpectraMax 190s. This information would be recorded as the model name under the part of the ontology that describes the equipment. Describing these readers in terms of the plate reader functionality part of the ontology enforces a record of the specifics of our laboratory setup. The <number of readings per well> is one, and there is one <measurement wavelength> at 595nm. The reader does <maintain internal temperature> at 30 degrees C. Usually the reader does not <agitate> the plate, as the plates are continually agitated while they are in the incubators but there is one occasion on which the reader must <agitate> <before measurement> for a <duration> of 30 seconds. This is to resuspend the yeast after centrifugation. This particular model of plate reader does not inform us what the <intensity> or <agitation mode> are. All this metadata is to be recorded in our database.

The next section describes the use of the ontology in the design of this database.

3.3 Use of the ontology for the design of a data model

As described above, a Robot Scientist will generate a very large amount of data and metadata. To ensure the integrity of this data, and to provide for its easy access, we will store all the data and metadata in a relational database.

The principal application of the ontology to database design was as an aid to identifying objects and events that needed to be recorded in the database. This was of key importance, as the primary aim when creating a good relational database design is to model the real world system as closely as possible. You first identify the objects and events that you want the database to represent: creating a structured ontology of your system is a good way of doing this. You then define the tables and all the relevant fields that they should contain, and finally describe how they are all related.

The ontology also helped with naming both tables and columns, with defining relationships between various data, and as a verification that the database design had incorporated all of the data useful to the project.

The fragment of the database design shown in Figure 7 handles the data records of individual 96-well plates; what model of plate and lid it is, what use it is being put to, what actions have happened to it during its lifetime within the Robot Scientist project, and the details of the robotic equipment that have been used to handle it. For each piece of equipment (e.g. a plate reader) it stores what settings were used and over what timeframe. This allows you to retrieve exactly what settings were used on any piece of equipment that interacted with any particular plate at any time in the history of the project.

To explain the main 'plates' table columns in more detail:

- **id_plate_barcode**: Each physical 96-well plate has a unique 8-digit <barcode> label attached to it for tracking purposes. There are three barcode readers on the Robot Scientist, one for each of the three subsystems (see Fig. 1). The plate is scanned once in the first subsystem to *create* it, and again on entry into subsystems two and three to check its identity before it is worked on. For example, 00012345.
- **id_plate_location**: Each physical position on the Robot Scientist where a plate can be placed or moved to has a unique <location> number, with all valid locations stored on the separate 'locations' table. For example the plate reader in subsystem three is location 3300.
- **id_plate_use_type**: This is a reference to the <plate usage> for the specific plate. These are held on the 'plate_use_types' table. There are currently three uses a plate could be put to; as a <yeast strain library plate>, as a <yeast pregrowth plate>, and as an <experiment nutrient cocktail plate>.
- **id_plate_status**: Each plate has a <status> associated with it to record its current condition. Generally a plate will initially start off in an <empty> state, then become <in use>, and then when it is finished with and disposed of it becomes <destroyed>. This allows us to quickly identify which plates are active and which are historical.
- **id_plate_model**: This is a reference to the <model> of plate, we use different models for different parts of the system; for example the yeast library plates are larger to accommodate greater volumes in deeper wells, whilst the experiment cocktail plates are made of clear polystyrene and have flat-bottomed wells to allow optical readings to be taken. Similar plates may also be made by different manufacturers so we need to record this. The various models of plate are stored on the 'plate_models' table which in turn is linked to supplier information (not shown).
- **is_plate_lidded**: A Boolean flag to indicate whether the plate has a <lid> or not.
- **id_lid_model**: This is a reference to the <lid model>. For example a lid may be flat or it may have ridges to reduce evaporation from wells. The various models of lid are stored on the 'lid_models' table.
- **created_ts**: This field is used to store the <timestamp> (time and date) of when the plate was created. In database terms this refers to the first time its unique barcode was scanned, normally when a robot arm has first taken it from a consumables plate stack for use.
- **created_by**: This field is used to store who or what created the plate. If the plate was manually created and introduced to the system (e.g. a yeast library plate) this field will contain the name of the person who set it up. Otherwise it will contain a name related to where on the Robot Scientist it was created.

As in the application of EXPO to curating and annotating experimental metadata, and the curation and annotation of metadata on experimental equipment, the application of a general experimental ontology to database design allows data and metadata to be compared and shared between experiments and laboratories.

4 DISCUSSIONS AND CONCLUSION

A Robot Scientist enables us to capture an unprecedented amount of information about scientific experiments. For the first time it is possible to completely capture and digitally curate all aspects of the scientific process. This presents us both with unique opportunities and challenges. The opportunity is the ability for the first time to record and fully understand how and why a particular experiment was conceived and executed, and to remove all subjectivity in experimental actions. This enables all aspects of experimentation, including hypothesis formation and testing, to be fully repeatable.

The great technical challenge is how to capture and digitally curate all this information. We argue that formation of a Robot Scientist ontology is a key step in meeting this challenge. We have used such an ontology to curate and annotate the experimental data and metadata and the equipment metadata, and to help design the associated database systems. As our ontology is linked to a general ontology of scientific experiments (EXPO) all the data and metadata captured can be shared with other experiments. We envisage our ontology as a start point for further community efforts in developing a general ontology for fully automated laboratories.

We believe that this increased ability to record and curate all aspects of scientific experiments will have important ramifications for scientific publishing. As in the e-Science 'vision' it will be increasingly easy to link papers to all the relevant data and metadata, ensuring full repeatability. In this task we believe that natural language will be required less and less to describe experiments. This is to be welcomed as natural language is notorious for its imprecision and ambiguity. Its use is also a great hindrance when using computers to store and analyse data—hence text-mining. We therefore argue that the content of scientific papers should increasingly be expressed in formal languages with ontological foundations.

ACKNOWLEDGEMENTS

We wish to thank all the members of the Robot Scientist team: Jem Rowland, Mike Young, Ken Whelan, Magdalena Markham, Emma Byrne and Wayne Aubrey. The work was funded by the BBSRC and the Royal Commission for the Great Exhibition of 1851.

REFERENCES

Bard,J. and Rhee,S. (2004) New ontologies in biology: design, applications and future challenges. *Nature Reviews. Genetics*, **5/3**, 213–222.

Brazma,A., Hingamp,P., Quackenbush,J., Sherlock,G. *et al*. (2001) Minimum information about a microarray experiment (MIAME):toward standards for microarray data. *Nature Genetics*, **29**, 365–371.

Clare,A. and King,R.D. (2003) Predicting gene function in *Saccharomyces cerevisiae*. In *Proceedings of the 2nd European Conference on Computational Biology*.

Guarino,N. (1998) Some ontological principles for designing upper level lexical resources. In *First International Conference on Language Resources and Evaluation (eds. Rubio, Gallardo, Castro, Tejada),* 527–534.

Jenkins,H., Hardy,N., Beckmann,M., Draper,J. *et al*. (2004) A proposed framework for the description of plant metabolomics experiments and their results. *Nature Biotechnology*, **22** (12), 1601–1606.

King,R.D., Whelan,K., Jones,M., Reiser,P. and Bryant, C. (2004) Functional genomics hypothesis generation by a robot scientist. *Nature*, **427, no 6971**, 247–252.

King,R., Young,M., Clare,A., Whelan,K. and Rowland,J. (2005) The robot scientist project. In *Springer Lecture Notes in Computer Science 3735 (eds. A.G. Hoffmann, Motada, H., Scheffer, T.),* pages 16–25.

Kozaki,K., Kitamura,Y., Ikeda,M. and Mizoguchi,R. (2002) Hozo: An environment for building/using ontologies based on a fundamental consideration of ''role'' and ''relationship''. In *Knowledge Engineering and Knowledge Management*, 213–218.

Masuda,M. and Ogur,M. (1969) Enzymatic and physiological properties of the yeast glutamate-alpha-ketoadipate transaminase. *J. Biol. Chem.*, **244**, 5153–8.

Mizoguchi,R. (2004) Tutorial on ontological engineering, Part 3: Advanced course of ontological engineering. *New Generation Computing*, **22/2**, 193–220.

Ryn,L. and Sternberg,P.W. (2003) Building a cell and anatomy ontology of *Caenorhabditis elegans*. *Comparative and Functional Genomics*, **4**, 121–126.

Soldatova,L. and King,R. (2005) Are the current ontologies used in biology good ontologies? *Nature Biotechnology*, **9/23**, 1096–1098.

Soldatova,L. and King,R. (2006) An ontology of scientific experiments. *Journal of the Royal Society Interface*, (in press).

Soldatova,L. (2005) EXPO. http://sourceforge.net/projects/expo/

Stoeckert,C., Causton,H. and Ball,C. (2002) Microarray databases: standards and ontologies. *Nature Genetics*, **32**, 469–473.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e472–e480
doi:10.1093/bioinformatics/btl250

ARTS: accurate recognition of transcription starts in human

Sören Sonnenburg[1], Alexander Zien[2,3] and Gunnar Rätsch[3,*]

[1]Fraunhofer Institute FIRST, Kekuléstr. 7, Berlin, Germany, [2]Max Planck Institute for Biological Cybernetics, Spemannstr. 38, Tübingen, Germany and [3]Friedrich Miescher Laboratory, Max Planck Society, Spemannstr. 39, Tübingen, Germany

ABSTRACT

We develop new methods for finding transcription start sites (TSS) of RNA Polymerase II binding genes in genomic DNA sequences. Employing Support Vector Machines with advanced sequence kernels, we achieve drastically higher prediction accuracies than state-of-the-art methods.

Motivation: One of the most important features of genomic DNA are the protein-coding genes. While it is of great value to identify those genes and the encoded proteins, it is also crucial to understand how their transcription is regulated. To this end one has to identify the corresponding promoters and the contained transcription factor binding sites. TSS finders can be used to locate potential promoters. They may also be used in combination with other signal and content detectors to resolve entire gene structures.

Results: We have developed a novel kernel based method – called *ARTS* – that accurately recognizes transcription start sites in human. The application of otherwise too computationally expensive Support Vector Machines was made possible due to the use of efficient training and evaluation techniques using *suffix tries*. In a carefully designed experimental study, we compare our TSS finder to state-of-the-art methods from the literature: *McPromoter*, *Eponine* and *FirstEF*. For given false positive rates within a reasonable range, we consistently achieve considerably higher true positive rates. For instance, *ARTS* finds about 35% true positives at a false positive rate of 1/1000, where the other methods find about a half (18%).

Availability: Datasets, model selection results, whole genome predictions, and additional experimental results are available at http://www.fml.tuebingen.mpg.de/raetsch/projects/arts

Contact: Gunnar.Raetsch@tuebingen.mpg.de

1 INTRODUCTION

Arguably the most important information about genomic DNA is the location of genes that encode proteins. For further analysis of the genes it is necessary to find their promoters and the contained binding sites of transcription factors, which are responsible for regulating the transcription of the gene.

Transcription start sites are located in the core promoter region and are usually determined by aligning complete mRNA or 5′-end EST sequences (for instance obtained by 5′ RACE) against the genome. Note that protein sequences and other ESTs are not sufficient for this task, since they typically start downstream of the TSS. For some species including human, large scale sequencing projects of complete mRNAs have been undertaken, but many low copy genes still evade being sequenced. In order to identify these genes and their promoter regions, computational TSS finding or better experimental techniques are the only way out.

Moreover, in the vast majority of species the identification of promoters must be accomplished without the support of massive sequencing. One possibility is to exploit homology to well-characterized genes in other species. While this approach can work for common genes, for those genes specific to some species or some family of species it is likely to fail. This leaves a huge demand for accurate *ab initio* TSS prediction algorithms.

Consequently, a fairly large number of TSS finders (TSF) has been developed. Generally TSFs exploit that the features of promoter regions and the TSS are different from features of other genomic DNA. Many different features have been used for the identification: the presence of CpG islands, specific transcription factor binding sites (TFBS), higher density of predicted TFBSs, statistical features of proximal and core promoter regions and homology with orthologous promoters (see Bajic *et al*., 2004; Werner, 2003) for two recent reviews on mammalian promoter recognition). Methods for recognizing TSSs employed neural networks, discriminant analysis, the Relevance Vector Machine (RVM), interpolated Markov models, and other statistical methods.

In a recent large scale comparison (Bajic *et al*., 2004;) eight TSFs have been compared. Among the most successful ones were *Eponine* (Down and Hubbard, 2002) (which trains RVMs to recognize a TATA-box motif in a G+C rich domain), *McPromoter* (Ohler *et al*., 2002) (based on Neural Networks, interpolated Markov models and physical properties of promoter regions) and *FirstEF* (Davuluri *et al*., 2001) (based on quadratic discriminant analysis of promoters, first exons and the first donor site, using CpG islands). *DragonGSF* (Bajic and Seah, 2003) performs similarly well as the aforementioned TSFs (Bajic *et al*., 2004). However, it uses additional binding site information based on the TRANSFAC data base (Matys *et al*., 2006); thus it exploits specific information that is typically not available for unknown promoters. For this reason and also because the program is currently not publicly available, we exclude it from our comparison.[1]

One characteristic of TSFs is that they normally rely on the combination of relatively weak features such as physical properties of the DNA or the G+C-content. In none of the above-mentioned approaches the recognition of the actual transcription start site has been seriously considered. In this work we show that by using very recently developed discriminative sequence analysis techniques (Sonnenburg

*To whom correspondence should be addressed.

[1] Further, unlike DragonGSF all of the above TSFs could – after retraining – be applied to genomes other than human, where only a few or no TF binding sites are known.

Fig. 1. Given two sequences $\mathbf{x}_1$ and $\mathbf{x}_2$ of equal length, the WD kernel with shift consists of a weighted sum to which each match in the sequences makes a contribution $\gamma_{k,p}$ depending on its length k and relative position p, where long matches at the same position contribute most significantly. The γ's can be computed from the β's and δ's in (2). The spectrum kernel is based on a similar idea, but it only considers substrings of a fixed length and the contributions are independent of the relative positions of the matches to each other.

et al., 2005) – which previously were only tractable on a much smaller scale (Rätsch *et al*., 2005) – we can drastically improve the performance of TSS recognition.

The remainder of the paper is structured as follows: In Section 2 we discuss the features of the sequences and kernels that we use for learning in order to recognize transcription start sites. In Section 3 we discuss techniques related to the kernels and Support Vector Machine training and evaluation, which were necessary in order to perform the experiments. We discuss the experimental setup and the data generation for a large scale comparison of our method *ARTS* with other TFSs, and provide experimental results in Sections 4 and 5. We conclude with a discussion and an outlook.

2 BASICS, FEATURES AND KERNELS

Binary classification methods aim at estimating a classification function $f : \mathcal{X} \rightarrow \{\pm 1\}$ using labeled training data from $\mathcal{X} \times \{\pm 1\}$ such that f will correctly classify most unseen examples (test data). In our case, the input space $\mathcal{X}$ will contain sequences $\{A, C, G, T\}^N$ centered at any genomic position, while the labels +1 or −1 indicate whether these positions are true TSS or decoy sites, respectively.

2.1 SVMs and Kernels

We use Support Vector Machines (Cortes and Vapnik, 1995) (SVMs) for two reasons. First, they exhibit a very competitive classification performance, since over-fitting is prevented by well-controllable regularization and training is not hampered by any local minima. Second, SVMs can conveniently be adapted to the problem at hand by designing appropriate kernel functions. The kernel function shortcuts mapping points to a feature space $\mathcal{F}$ via an arbitrary function Φ and computes dot products in that space via $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. Particularly advantageous for TSS recognition is the possibility to build complex modular kernel functions by combining several simpler ones. This way of combining different pieces of information has been shown to be very powerful (e.g. [10]).

For a test example $\mathbf{x}$ the classification function generated by an SVM can be written as

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b \right), \tag{1}$$

where $y_i \in \{\pm 1\}$ is the label of training example $\mathbf{x}_i$ $(i = 1, \ldots, N)$. The coefficients α_i and the bias b are the results of SVM training. Please note that (1) is equivalent to $f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b$, where $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \Phi(\mathbf{x}_i) \in F$. The SVM constructs a maximum margin linear classifier in the Φ-space. The computation of SVMs only depends on the inner products of training examples; therefore it is usually sufficient to specify the kernel function $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$ that computes the inner products in feature space.

2.2 Features for TSS recognition

As most other TSFs our method combines several features, thereby utilizing prior knowledge about the structure of transcription start sites. We put, however, particular care in analyzing the actual transcription start site. We have considered the following:

- The TSS is only determined up to a small number of base pairs. Further, nearby binding sites may also not be positionally fixed. In order to model the actual TSS site, we thus need a set of features that are approximately localized and allow for limited flexibility. We have recently proposed a kernel — the extended Weighted Degree kernel *with shifts* (WD_S) — for the identification of alternatively spliced exons [16] which is also well suited for this task:

$$k(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{K} \beta_k \sum_{l=1}^{L-k+1} \sum_{\substack{s=0 \\ s+l \leq L}}^{S} \delta_s \, \mu_{k,l,s,\mathbf{x},\mathbf{x}'},$$

$$\mu_{k,l,s,\mathbf{x},\mathbf{x}'} = \mathbf{I}(\mathbf{u}_{k,l+s}(\mathbf{x}) = \mathbf{u}_{k,l}(\mathbf{x}')) + \mathbf{I}(\mathbf{u}_{k,l}(\mathbf{x}) = \mathbf{u}_{k,l+s}(\mathbf{x}')), \tag{2}$$

  where $\beta_k = 2(K - k + 1)/(K(K + 1))$, $\delta_s = 1/(2(s + 1))$ and $\mathbf{u}_{k,l}(\mathbf{x})$ is the subsequence of $\mathbf{x}$ of length k that starts at position l. The idea is to count the matches between two sequences $\mathbf{x}$ and $\mathbf{x}'$ between the words $\mathbf{u}_{k,i}(\mathbf{x})$ and $\mathbf{u}_{k,i}(\mathbf{x}')$ where $\mathbf{u}_{k,i}(\mathbf{x}) = x_i x_{i+1} \ldots x_{i+k-1}$ for all i and $1 \leq k \leq K$. The parameter k denotes the length of the words to be compared, and S is the maximum distance by which a sequence is shifted. See Figure 1 and [16] for details.
- Upstream of the TSS lies the promoter, which contains transcription factor binding sites. Comparing different promoters, it was noted that the order of TFBS can differ quite drastically. Thus, we use the so-called spectrum kernel [11] on a few hundred bps upstream of the TSS. The spectrum kernel is typically used to recognize regions in which certain k-mers are over- or under-represented (''content sensors''):

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\sigma \in \Sigma^k} \#\{\sigma \text{ appears in } \mathbf{x}\} \cdot \#\{\sigma \text{ appears in } \mathbf{x}'\},$$

  where $\#\{\sigma \text{ appears in } \mathbf{x}\}$ is the number of times a k-mer σ appears as a substring in $\mathbf{x}$. Since it does not preserve the information where the subsequences are located, it may not be appropriate for modeling localized signal sequences such as the actual transcription start site.
- Downstream of the TSS follows the 5′ UTR, and further downstream introns and coding regions. Since these sequences may significantly differ in oligo-nucleotide composition from intergenic or other regions, we use a second spectrum kernel for the downstream region.

Table 1. Parameters of the combined kernels and the SVM for TSS recognition. The ranges are specified according to our prior knowledge or intuition. A parameter value of 0 marked with * means that the sub-kernel is excluded from the combined kernel

Parameter	Set of values	Init. guess	Opt. value	Explanation
TSS signal (weighted degree with shift):				
• r-start	$\{-100, -90, \ldots, -10\}$	-50	-70	start of considered sequence region
• r-end	$\{+10, +20, \ldots, +100\}$	$+50$	$+70$	end of considered sequence region
• order	$\{0^*, 2, \ldots, 24\}$	10	24	length of substrings compared
• shift	$\{4, 8, \ldots, 48\}$	20	32	positional shift (base pairs)
Promoter (spectrum):				
• r-start	$\{-1000, -900, \ldots, -100\} \cup \{-150\}$	-600	-600	start of considered sequence region
• r-end	$\{-200, -150, \ldots, +200\}$	0	0	end of considered sequence region
• order	$\{0^*, 1, \ldots, 6\}$	3	4	length of substrings considered
1st exon (spectrum):				
• r-start	$\{-100, -50, \ldots, +300\}$	$+100$	0	start of considered sequence region
• r-end	$\{+100, +200, \ldots, +1000\}$	$+600$	$+900$	end of considered sequence region
• order	$\{0^*, 1, \ldots, 6\}$	3	4	length of substrings considered
angles (linear):				
• r-start	$\{-1000, -900, \ldots, -200\}$	-600	-600	start of considered sequence region
• r-end	$\{-600, -500, \ldots, +200\}$	-100	-100	end of considered sequence region
• smoothing	$\{0^*, 10, \ldots, 100\}$	50	70	width of smoothing window
Energies (linear):				
• r-start	$\{-1000, -900, \ldots, -200\}$	-600	–	start of considered sequence region
• r-end	$\{-600, -500, \ldots, +200\}$	-100	–	end of considered sequence region
• smoothing	$\{0^*, 10, \ldots, 100\}$	50	0^*	width of smoothing window
SVM: • C	$\{2^{-2.5}, 2^{-2}, \ldots, 2^{+2.5}\}$	2^0	2^1	regularization constant

- The 3D structure of the DNA near the TSS must allow the transcription factors to bind to the promoter region and the transcription to be started. To implement this insight, we apply two linear kernels to the sequence of twisting angles and stacking energies. Both properties are assigned based on dinucleotides as done by the *emboss* program *btwisted*.[2] The fourth and fifth kernel are then computed as the inner product $k(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}'$, where $\mathbf{x}$ is derived from a sequence of DNA twisting angles and stacking energies, respectively, by smoothing with a sliding window and using only every 20th of the resulting values.

The combined kernel is simply the sum of all sub-kernels, which is equivalent to appending the feature vectors in feature space. The sub-kernels can be expected to be of different importance for the overall performance; thus, it may seem appropriate to use a weighted sum. Experiments to verify this (not shown) indicated that a uniform weighting performs just as well as reducing the weights for the less important sub-kernels. An explanation for this may be that the SVM is able to learn relative weights itself. The only requirement is that the (weighted) function values of the sub-kernels are on a comparable scale; otherwise, those on a low scale are effectively switched off.

Note that we normalized all kernels with the exception of the linear kernels such that the vectors $\Phi(\mathbf{x})$ in feature space have unit length. This can be done efficiently by redefining the kernel as follows:

$$\tilde{k}(\mathbf{x}, \mathbf{x}') = \frac{k(\mathbf{x}, \mathbf{x}')}{\sqrt{k(\mathbf{x}, \mathbf{x}) k(\mathbf{x}', \mathbf{x}')}}. \quad (3)$$

[2] http://www.hgmp.mrc.ac.uk/Software/EMBOSS/Apps/btwisted.html

This normalization solves convergence problems of SVM optimizers and balances the importance of the kernels among each other. In total we combine five kernels which each have several parameters as listed in Table 1.

3 EFFICIENCY CONSIDERATIONS

Our model is complex in that it consists of several sophisticated kernels applied to rather long stretches of DNA. Furthermore, we have to train it on as many examples as possible in order to attain a high prediction accuracy.[3] Even with highly optimized general purpose SVM packages like *LibSVM* or *SVM*^{light}, training and tuning our model with tens of thousands of points is intractable. The main reason is that the kernel computation, in particular of the WD kernel with shift, is very expensive (single kernel computation $\mathcal{O}(KLS)$). In addition, many kernel elements need to be computed several times when the kernel cache is not large enough, which is quite likely with $\gg$10,000 examples. However, fast training is possible without kernel caching, if the SVM output for any training point $\mathbf{x}_i$, i.e. $\mathbf{w} \cdot \Phi(\mathbf{x}_i)$, can be computed efficiently. In the following subsection, we show how algorithms can be modified to take advantage of fast computations of $\mathbf{w} \cdot \Phi(\mathbf{x}_i)$ during training and testing. In the second subsection we show how this can be accomplished for the different kernels that we use.

[3] For instance on a splice site recognition task we were able to reduce the error rate by 20% when doubling the amount of training data – over a wide range of training set sizes (Sonnenburg *et al*., 2006).

Algorithm 1 Outline of the decomposition algorithm that exploits the fast computations of linear combinations of kernels (e.g. by suffix tries).

$f_i = 0$, $\alpha_i = 0$ for $i = 1, \ldots, N$
for $t = 1, 2, \ldots$ **do**
  Check optimality conditions and stop if optimal
  select Q variables $i_1, \ldots, i_Q$ based on $\mathbf{f}$ and α
  $\alpha^{old} = \alpha$
  solve SVM dual w.r.t. the selected variables and update α
  generate data structures to prepare efficient computation of $g(\mathbf{x}) = \sum_{q=1}^{Q} (\alpha_{i_q} - \alpha_{i_q}^{old}) y_{i_q} k(\mathbf{x}_{iq}, \mathbf{x})$
  update $f_i = f_i + g(\mathbf{x}_i)$ for all $i = 1, \ldots, N$
end for

3.1 Faster SVM training and evaluation

As it is not feasible to use standard optimization toolboxes for solving large scale SVM training problems, decomposition (also called *chunking* in the machine learning literature) techniques are used in practice. Most decomposition algorithms work by first selecting a working set $W \subseteq \{1, \ldots, N\}$ with (the indices of) Q variables of the N training points based on the current solution. Then the corresponding reduced problem is solved with respect to the working set variables. These two steps are repeated until some optimality conditions are satisfied [see e.g. Joachims (1998)].

Efficient Updates in Decomposition Algorithms For selecting the working set and checking the termination criteria in each iteration, the vector $\mathbf{f}$ with $f_i = \sum_{j=1}^{N} \alpha_j y_j k(\mathbf{x}_i, \mathbf{x}_j), i = 1, \ldots, N$, is needed. To avoid computation of $\mathbf{f}$ in every iteration one typically starts with $\mathbf{f} = 0$ and computes updates of $\mathbf{f}$ on the changed variables:

$$f_i \leftarrow f_i^{old} + \sum_{j \in W} (\alpha_j - \alpha_j^{old}) y_j k(x_i, x_j), \quad \forall i = 1, \ldots, N,$$

where $Q = |W|$ is the size of the working set. One typically uses kernel-caching to reduce the computational effort of this operation, which is, however, is not sufficient in the case of large training sets. Fortunately, for all kernels considered in this work we can efficiently compute linear combinations of kernel elements, i.e. $\mathbf{w} \cdot \Phi(\mathbf{x})$, where $\mathbf{w}$ is of the form $\sum_i \alpha_i y_i \Phi(\mathbf{x}_i)$. Algorithm 1 implements a simple idea how to use this to speedup SVM training.

For all considered kernels, the operation $\mathbf{w} \cdot \Phi(\mathbf{x})$ is almost as cheap as computing the dot product of two $\Phi(\mathbf{x})$'s. Hence, Algorithm 1 leads to a speedup of up to factor Q. Note that creating the data structure for Q examples (e.g. the below-mentioned suffix tries) can be expensive, however, it is a fixed cost per iteration. If the number of examples is large enough, then the speedup of the evaluation leads to a great advantage.

Efficient Evaluation In the application we have in mind we need to compute predictions for every position in the human genome ($\approx 7 \cdot 10^9$). For kernel methods that generate several thousands of support vectors, each of which is of length one thousand, this would mean more than 10^{16} floating point operations. This is too much even for modern computer cluster systems. By using the same idea as in training we can efficiently compute the SVM prediction $f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x})$ for new sequences. This leads to a speedup of M, where M is the number of Support Vectors (with $\alpha_i \neq 0$), and makes the genome-wide computation of promoter predictions feasible — with still ≈350 h computing time for the entire human genome.

3.2. Fast string kernel computations

All considered kernels correspond to a feature space $\mathcal{F}$ that can be very high dimensional. For instance in the case of the WD kernel on DNA sequences of length 100 with $K = 20$, the corresponding feature space is 10^{14} dimensional (one feature per position and possible k-mer, $1 \leq k \leq K$). However, most dimensions in the feature space are not used since only a few of the many different k-mers actually appear in the sequences. An appropriate choice of the data representation is therefore crucial for fast algorithms. If the data can be efficiently represented as sparse vectors in the feature space $\mathcal{F}$, one achieves significant speedups in SVM training and testing.

Spectrum kernels with explicit feature maps If the dimensionality of the feature space is small enough, then one can store the whole vector $\mathbf{v} \in \mathcal{F}$ in memory and perform direct operations on its elements. This is true for our linear kernels (4 and 5) and also the spectrum kernel for relatively short K-mers (e.g. $K = 6$ leads to a 4096 dimensional space). For the latter case one may first preprocess the sequences $\mathbf{x}$ into a sparse vector $\Phi(\mathbf{x})$ and later perform computations with mixed sparse and full vectors, which can be implemented very efficiently. This approach has exponentially growing memory demands ($\mathcal{O}(|\Sigma|^K)$), but is very fast and best suited for instance for the spectrum kernel on DNA sequences with $K \leq 14$ and on protein sequences with $K \leq 6$.

WD kernels with suffix tries The difference between the WD kernel (without shifts) and the spectrum kernel is (a) the position dependence and (b) the consideration of K-mers vs. $1, \ldots, K$-mers, i.e. also including subsequences of the K-mers. If one would use a weighted sum of spectrum kernels for all degrees $\leq K$ at every position of the sequence, then it is equivalent to the WD kernel (Sonnenburg *et al*., 2005). So in principle we could apply the idea used for the Spectrum kernel to speedup the WD kernel as well. However, when using long K-mers (e.g. $K = 20$) the memory demand becomes intractable and the sparse weight vectors need to be stored and operated with more efficiently. In (Sonnenburg *et al*., 2006) we have suggested to use *suffix tries*, i.e. trees that store weights not only at the leaves but also at internal nodes. The idea is to use one trie of degree four ($|\Sigma|$) and depth K per position in the sequence. A node in the trie at depth k is addressed by a k-mer and stores its associated value. See Figure 2 for illustration. Note that we can easily add several sequences to the trie and the worst-case cost for performing a lookup operation is $\mathcal{O}(K)$. This is the key to the speedup of SVM training and evaluation.

Please note that the tries for the WD kernel *with shifts* can be analogously constructed. Now a string has to be found several times. Either we store it in several neighboring trees (with decaying weights) or we store it only once and query the neighboring trees during lookup operations. So far the latter version was used, which turned out to require too much computing time. The first option, however, requires rather large tries as each string is stored in several tries. This particularly matters during testing when the trie needs to store all subsequences of support vectors and one tree can grow to more than 200Mb. One therefore cannot build all trees at once, but only sequentially. For considerations of how to extend this approach to mismatching k-mers see (Sonnenburg *et al*., 2005).

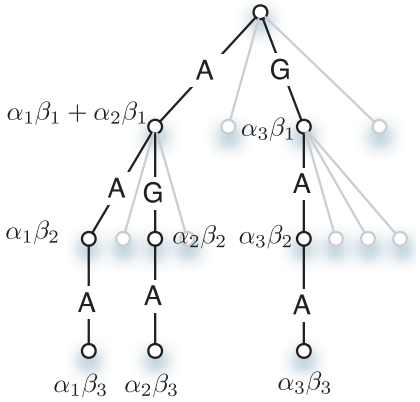


Fig. 2. Three sequences AAA, AGA, GAA being added to the trie. The figure displays the resulting weights at the nodes, where the β's correspond to a weighting over the the depth of the trie and the α's are the sequence weights.

Implementations The implementation of the kernel functions, their efficient computation during SVM training (including Multiple Kernel Learning (Sonnenburg *et al*., 2006)) and evaluation is implemented in *C++* and will be made available as part of a kernel learning toolbox called *Shogun* interfacing to *R*, *Octave*, *Matlab* and *Python* (see http://www.fml.mpg.de/raetsch/projects/shogun).

4 TRAINING AND TUNING THE MODEL

For training a TSF and selecting its model parameters (''model selection'') it is crucial to use proper training and testing data. In Section 4.1, we elaborate on the generation of suitable datasets, including all relevant steps of data pre-processing. We also explain in necessary detail how we perform the model selection (Section 4.3).

4.1 Datasets

Both for training our TSS finder and for assessing its accuracy we need known TSSs as well as known non-TSSs.

To generate TSS data for training, we use dbTSS (Suzuki *et al*., 2002) version 4 (''dbTSSv4''), which is based on the human genome sequence and annotation version 16 (''hg16''). It contains transcription start sites of 12763 RefSeq genes (Kim *et al*., 2005). First we extract RefSeq identifiers from dbTSSv4 and then obtain the corresponding mRNA sequences using NCBI nucleotide batch retrieval.[4] Next, we align these mRNAs to the hg16 genome using BLAT [9].[5] From dbTSS we extracted putative TSS positions (Field: Position of TSS) which we compared with the best alignment of the mRNA. We discard all positions that do not pass all of the following checks: 1. Chromosome and strand of the TSS position and of the best BLAT hit match. 2. The TSS position is within 100 base pairs from the gene start as found by the BLAT alignment. 3. No already processed putative TSS is within 100bp of the current one. This procedure leaves us with 8508 genes, each annotated with gene start and end. To generate positive training data, we extract windows of size $[-1000, +1000]$ around the TSS.

To discriminatively train a classifier one also needs to generate ''negative'' data. However there is no single natural way of doing this: since there are further yet unknown TSS hidden in the rest of the genome, it is dangerous to sample negative points randomly from it. So we choose to proceed similarly to Bajic *et al*., (2004) by extracting ''negative'' points (again, windows of size $[-1000, +1000]$) from the interior of the gene. More precisely, we draw 10 negatives at random from locations between 100 bp downstream of the TSS and the end of the gene.[6] We finally obtain 8508 positive and 85042 negative examples, of which we will use 50% for training a TSS classifier and 50% for validating it. The final evaluation is done on a differently generated test data set (cf. Section 5.2).

4.2 Performance measures

We use two established measures of performance as guidance for model selection and, later on, for evaluating our success. The sensitivity (or recall) is defined as the fraction of correctly classified positive examples among the total number of positive examples, i.e. it equals the true positive rate $TPR = TP/(TP + FN)$. Analogously, the fraction $FPR = FP/(TN + FP)$ of negative examples wrongly classified positive is called the false positive rate. Plotting FPR against TPR results in the Receiver Operator Characteristic Curve (ROC) (Metz, 1978; Fawcett, 2003). Plotting the true positive rate against the positive predictive value (also precision) $PPV = TP/(FP + TP)$, i.e. the fraction of correct positive predictions among all positively predicted examples, one obtains the Precision Recall Curve (PRC) (see e.g. [4]). For both graphs, the area under the curve is a useful single-number performance measure, which we refer to as *auROC* (for area under ROC) and *auPRC*, respectively.

4.3 Model selection

As seen before (Table 1), there are many (in fact, 17) parameters that need to be set to reasonable values in order for our approach to work well. We treat this as a model selection problem: each parameter setting corresponds to a set of assumptions, i.e. a model, on what distinguishes the surroundings of TSS from other genomic loci. We want to select the closest approximation (within the framework defined by the kernel function) to reality, which can be identified by having the best predictive power. Thus we train the SVM with different parameter settings and assess the resulting prediction performance on a separate validation set.

While model selection is often done by trying all points on a regular grid in the space of parameters, this is computationally infeasible for more than a few parameters. Therefore, we resort to iterated independent axis-parallel searches. First, we specify a start point in parameter space based on prior knowledge and intuition. Then, in each round candidate points are generated by changing any single parameter to any value from a pre-defined small set; this is done for every parameter independently. Finally, the new parameter setting is assembled by choosing for each parameter the value that performed best while leaving the other parameter values unchanged.

We choose the model that yields the highest auROC on the validation set. It achieves 93.99% auROC and 78.20% auPRC

[4] http://ncbi.nih.gov/entrez/batchentrez.cgi?db=Nucleotide

[5] We used the options `-tileSize=16 -minScore=100 -minMatch=4 -minIdentity=98 -t=dna -q=rna`.

[6] If a gene is too short, fewer or even no negative examples are extracted from that particular gene.

Table 2. Results obtained by removing sub-kernels. The energies kernel is already turned off by the model selection

Subkernel	Area under ROC	Area under PRC
w/o TSS signal	**90.75%**	**70.72%**
w/o promoter	93.33%	74.94%
w/o 1^{st} exon	92.76%	74.94%
w/o angles	93.99%	78.26%
Complete	93.99%	78.20%

Table 3. Results obtained when only a single specific sub-kernel is used. The actual TSS signal discriminates strongest, but also the 1^{st} exon carries much discriminative information

Subkernel	Area under ROC	Area under PRC
TSS signal	**91.42%**	**69.38%**
Promoter	86.55%	55.33%
1^{st} exon	88.54%	64.29%
angles	45.31%	7.86%

(99.93% and 99.91% on the training data, respectively). The selected parameter settings are shown in the second but last column of Table 1.

4.4 Importance of the kernels

In addition to optimizing the parameter settings of all sub-kernels, we investigate whether and how much each sub-kernel contributes to the overall classification. To do so, we remove each sub-kernel and retrain the remaining model (with all other parameters kept fixed at the selected values). The accuracies obtained on the validation set are shown in Table 2. Removing the WD_S kernel, which models the signal at $[-70, +70]$ around the TSS, decreases the performance of the classifier most, although it still performs rather well (auROC > 90%). The 1^{st} exon kernel, which models the 4-mer nucleotide frequency in the range $[0, +900]$ downstream, appears to be of second most importance in our kernel ensemble. Removing the linear kernels, which take into account the binding energies and the twisting of the DNA, has almost no effect on the result.

A different view on the contribution of the individual kernels can be obtained by retraining single-kernel SVMs. The respective results are displayed in Table 3. Again the WD_S kernel contributes most, followed by the two spectrum kernels modeling the first exon and the promoter. The DNA twistedness angle-measure performs even worse than at random, probably because SVM's regularization parameter C was not properly tuned for the single kernel case.

For illustration we analyze in Figure 3 how the TSS signal predictions are localized relative to the true transcription start sites. We consider a window of ±1000 around a true TSS and record the location of the maximal TSS signal prediction (TSS signal kernel only). Figure 3 displays a histogram of the recorded positions on our validation set. We observe an expected strong concentration near the true TSS. We also observe that the distribution is skewed – a possible explanation for this is offered by Figure 4:

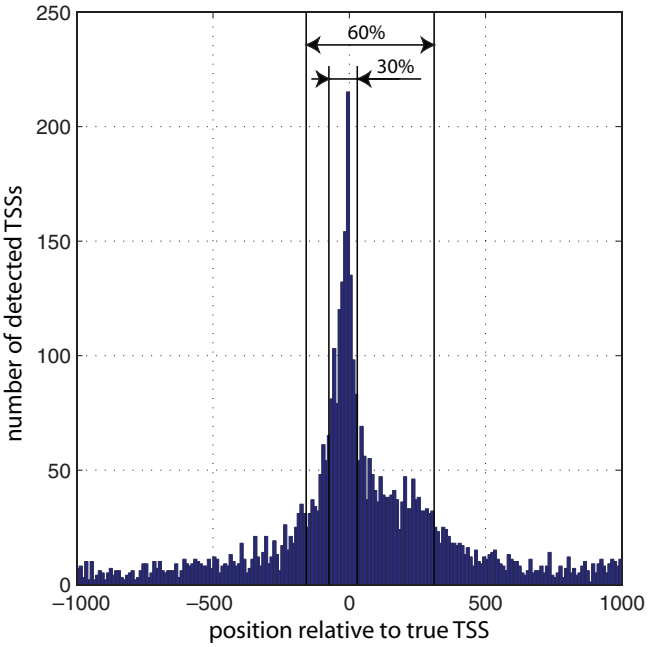


Fig. 3. Localization of *ARTS*'s TSS signal predictions: Shown is a histogram over the location with maximal score in a window ±1000 around true TSSs. In 60 % of the cases the predicted TSSs is within [−150, + 270]bp of the true TSS (30 % within [−70, + 40]).

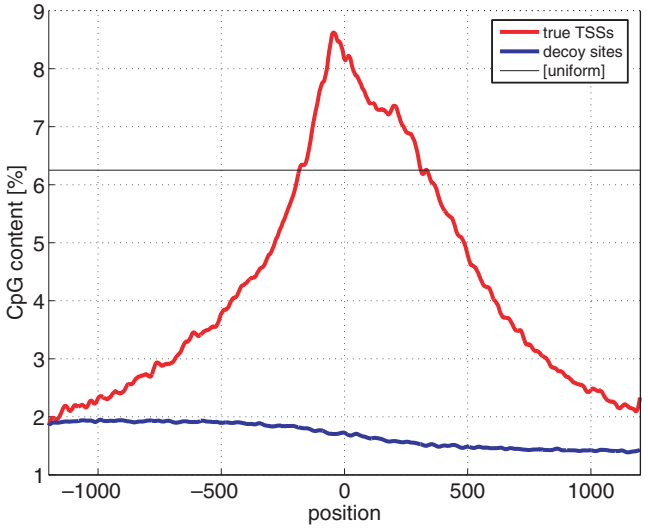


Fig. 4. Average positional frequency of CpG dinucleotides around true TSS and in decoy sequences (smoothed by convolution with a triangle of 39 bps length).

the predictor might be mislead by the distribution of CpG islands, which is skewed in a similar manner.

In conclusion it seems to be the WD_S kernel that models the region around the TSS best. The relatively large shift of 32 found by the model selection suggests the existence of motifs located around the TSS at highly variable positions. Neither *eponine* nor *FirstEF* model this regions explicitly. Thus, the WD_S kernel seems to be one of the reasons for ARTS' superior accuracy.

5 RESULTS AND DISCUSSION

We compare the performance of *ARTS*, our proposed TSS finding method, to that of *McPromoter* (Ohler *et al*., 2002), *Eponine* (Down and Hubbard, 2002) and *FirstEF* (Davuluri *et al*., 2001), which are among the best previously reported methods (Bajic *et al*., 2004). The evaluation protocol highly affects a TSF comparison; we thus give a detailed explanation (Section 5.1) of the criteria we use.

5.1 Setup

As POL II binds to a rather vague region, there seems to be no single true TSS location, but rather regions of roughly [−20, +20] bp constituting potential TSSs. For that reason one has to use evaluation criteria different from the ones used in standard two-class-classification. Bajic *et al*. (2004) suggest to cluster predicted TSS locations that have at most 1000bp distance to the neighboring locations. As evaluation criterion for each gene, they score a true positive if a prediction is located within ±2000 bp of the true TSS (otherwise, a false negative is counted); false positives and true negatives are counted from the TSS position +2001 to the end of the gene. However, each TSF is tuned to obtain a maximum true positive rate at a *different* false positive rate. Hence, this criterion suffers from the fact that it remains unclear how to compare results when the sensitivity and positive predictive value are both different (cf. Table 2 in Bajic *et al*. (2004)).

To alleviate this problem and allow for direct comparison via Receiver Operator Characteristic and Precision Recall Curves (ROC and PRC) we propose a different evaluation criterion. We compute whole genome point-wise predictions, which are then converted into *non-overlapping* fixed length chunks (e.g. of size 50 or 500). Within each chunk the maximum TSF output is taken. One can think of this chunking[7] process as a ''lens'', allowing us to look at the genome at a lower resolution. Obviously, for a chunk size of 1 this is the same as a point-wise TSS prediction. As ''lenses'' we use chunk sizes of 50 and 500. A chunk is labeled as +1 if it falls within the range ± 20 bp of an annotated TSS; chunks downstream of this range until the end of the gene are labeled −1.

Note that according to the above scheme some TSS will label two chunks as positive. This, however, does not constitute a problem, as it is a rare event if the chunk size is large. Furthermore, it is not unlikely that a TSF predicts both chunks as positive, as the maximum of the scores within each chunk is taken. We also considered an alternative criterion, in which only the chunk in which the maximum TSFs output is larger is labeled as positive, whereas the other chunk is removed from the evaluation. As a downside, this introduces a labeling that is dependent on the TSF output (i.e. there is no ground truth labeling over all TSFs), and leads to only small variations (auROC/auPPV increased/decreased by $\leq$3.5% for chunk size 50 and $\leq$1% for *all* TSFs for chunk size 500). Chunks obtain negative labels if they were not positively labeled and fall within the range gene start +20 bp to gene end and are excluded from evaluation otherwise.

This way the labeling of the genome stays the same for all TSFs. Considering *all* TSSs in dbTSSv5 we obtain labelings for chunk size 50 (500) with 28,366 (16,892) positives and 16,593,892 (1,658,483) negatives where TSS fall into two chunks in 15,223 (1,499) cases, covering in total 829,694,600 bp (≈12%) of the human genome.[8] In summary, the chunking allows for a controlled amount of positional deviations in the predictions. Unlike the clustering of predictions, it does not complicate the evaluation or hamper the comparability of TSF.

Table 4. Evaluation of the Transcriptions Start Finder at a chunk size resolution of 50 on dbTSSv5 excluding dbTSSv4 using the area under the Receiver Operator Characteristic Curve and the area under the Recall Precision Curve (larger values are better). For details see text

dbTSSv5-dbTSSv4 evaluation on chunk size 50		
TSF	Area under ROC	Area under PRC
Eponine	88.48%	11.79%
McPromoter	**92.55%**	6.32%
FirstEF	71.29%	6.54%
ARTS	**92.77%**	**26.18%**

Table 5. Evaluation of the Transcriptions Start Finder at a chunk size resolution of 500 on dbTSSv5 excluding dbTSSv4 using the area under the Receiver Operator Characteristic Curve and the area under the Recall Precision Curve (larger values are better). For details see text

dbTSSv5-dbTSSv4 evaluation on chunk size 500		
TSF	Area under ROC	Area under PRC
Eponine	91.51%	40.80%
McPromoter	**93.59%**	24.23%
FirstEF	90.25%	40.89%
ARTS	**93.44%**	**57.19%**

5.2 Test dataset

To allow for a fair comparison of promoter detectors, one needs to create a proper test set such that no promoter detector has seen the examples in training. We decide to take all ''new'' genes from dbTSSv5 (Yamashita *et al*., 2006) (which is based on hg17) for which a representative TSS was identified (i.e., the field ''The selected representative TSS'' is not empty). From dbTSSv5 we remove all genes that already appear in dbTSSv4 according to the RefSeq NM identifier. To take care of cases in which IDs changed over time or are not unique, we also remove all genes from dbTSSv5 for which mRNAs overlap by more than 30%. This leads to a total of 1,024 TSS to be used in a comparative evaluation. The comparison is done on this test set using chunk sizes 50 and 500 as resolutions (cf. Section 5.1), which results in 1,588 (943) positives and 1,087,664 (108,783) negatives. In 816 (67) cases the TSS fall into two chunks.

5.3 TSF Performance evaluation

ROC curves are an established criterion for comparing classifiers. While they are meaningful on balanced datasets, they lose explanatory value when highly skewed datasets are compared. Exactly

[7] Not to be confused with the ''chunking'' (decomposition) algorithms used for SVM training.

[8] Here we used the dbTSS field *Position(s) of 5'end(s) of NM_(or known transcript)* as the field *Selected representative TSS* is often empty.

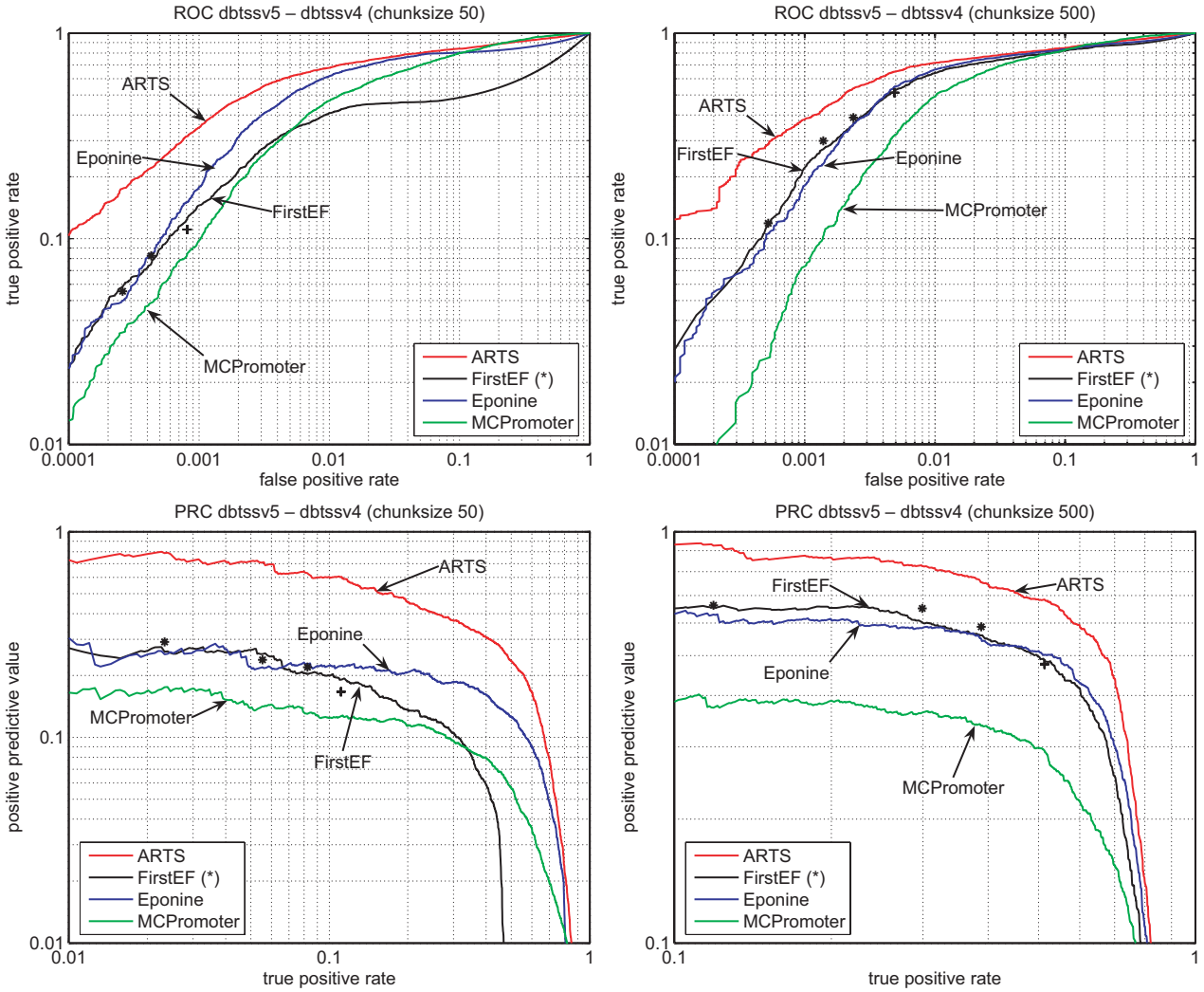


Fig. 5. Performance evaluation of the *ARTS, FirstEF, Eponine* and *McPromoter* TSS predictors. Evaluation was done on all genes whose TSS was found to be newly annotated in dbTSSv5 (i.e. genes whose TSS was not already in dbTSSv4). Receiver Operator Characteristic and Precision Recall Curves on decreased output resolution were computed (taking the maximum output within non-overlapping chunks of size 50 (left column) and 500 (right column for more details see text). Windows were marked positive if a known TSS lies in a range of ±20bp and negative otherwise. Please note that the 'bumps' in the upper right corner in the FirstEF/Eponine plots for low window sizes are artifacts, caused by the method not giving predictions for every position. However the interesting area is in the left (lower false positive rate).

this is the case for TSS prediction, where we have about 7 billion loci of which, even for optimistic guesses, less than 3 million bps, i.e. only 0.05%, belong to true TSS. Let us consider a TSF that correctly classifies 100% of the true TSS sites (TPR) while wrongly classifying 1% of the non TSS loci (FPR). The area under the ROC would score at least 99% suggesting a particularly good classifier. However if only 0.05% of the negative examples (which in absolute values is 300 million) achieve a higher score than all of the positive examples (3 million), the area under the Precision Recall Curve will be less than 1%. For a reliably useful measure of prediction accuracy, we thus resort to the auPRC.

As the performance evaluation via ROC/PRC curve needs (genome-wide) real valued outputs for each TSF, we set the TSF's thresholds to the lowest acceptable values. *Eponine* is run with the options -threshold 0.5. As *McPromoter* provides the outputs for every tenth base pair we can use the unfiltered raw values directly. *FirstEF* does not provide a single score as output, but probability scores for the promoter, exon, and donor. By default, a prediction is made if each probability equals or is larger than a pre-defined threshold (promoter: 0.4, exon: 0.5, donor 0.4), which yields just a single point in the ROC and PRC space. We therefore set all thresholds to 0.001 and later use the product of the scores as a single output.[9]

Next we chunk the output, as described above in Section 5.1, and perform evaluation on all genes whose TSS was found to be newly annotated in dbTSSv5 (cf. Section 5.2). Tables 4 and 5 display the results for the performance measures area under the ROC and PRC curve for *ARTS*, *FirstEF*, *McPromoter*, and *Eponine*. Table 4 shows

[9] As a validation we run *FirstEF* with the default settings and a variety of other thresholds. The obtained TPR/FPR and TPR/PPV values fit the curves produced using the single score extremely well (cf. Figure 5 below).

results for chunk size 50 and Table 5 for chunk size 500, corresponding to different levels of positional accuracy or resolution. In both cases our proposed TSS finder, *ARTS*, clearly outperforms the other methods in terms of both auROC and auPRC. This is also seen in Figure 5, which supplies detailed information on the true positive rates (top) and the positive predictive values (bottom) for a range of relevant true positive rates.

An interesting observation is that, judging by the auROC, McPromoter constitutes the second best performing TSF, while, on the other hand, it performs worst in the auPRC evaluation. An explanation can be found when looking at the ROC and PRC in Figure 5 where the left column displays ROC/PRC for chunk size 50 and the right for chunk size 500. Analyzing the ROC figures, we observe that *McPromoter* outperforms *FirstEF* and *Eponine* for false positive rates starting around 10% – a region contributing most to the auROC (note that both axes are on log scale). All of the three aforementioned promoter detectors perform similarly well. At a reasonable false positive level of 0.1% the TSFs perform as follows (chunk size 50): *ARTS* 34.7%, *Eponine* 17.9%, *FirstEF* 14.5%, and McPromoter 9.8%. Also note that the ROC curves for both chunk sizes are very similar, as the ROC curves are independent of class ratios between the negative and the positive class.[10] On the other hand class ratios affect the PRC quite significantly: For instance, at a true positive rate of 50%, *ARTS* achieves a PPV of 23.5% for chunk size 50 and 68.3% for chunk size 500. *ARTS*'s ROC and PRC, however, constantly remains well above its competitors.

6 CONCLUSION

We have developed a novel and *accurate* transcription start finder, called *ARTS*, for the human genome. It is based on Support Vector Machines that previously were computationally too expensive to solve this task. It has therefore been an important part of our work to develop more efficient SVM training and evaluation algorithms using sophisticated string kernels. In a carefully designed experimental study we compared *ARTS* to other state-of-the-art transcription start finders that are used to annotate TSSs in the human genome. We show that *ARTS by far* outperforms all other methods: it achieves true positive rates that are twice as large as those of established methods. In the future, we plan to train and evaluate the *ARTS* system on other genomes and make the system as well as its predictions publicly available. It would be interesting to see how much of *ARTS*' higher accuracy can be translated into improved *ab initio* gene predictions.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge partial support from the PASCAL Network of Excellence (EU #506778), DFG grants JA 379/13-2 and MU 987/2-1. We thank Uwe Ohler, Vladimir B. Bajic, Michael Zhang, Ramana Davuluri, Ivo Grosse, Koji Tsuda, and Klaus Robert Müller for helpful and motivating discussions. We thank P. Philips, F. de Bona, C. Dieterich, and G. Zeller for proofreading the manuscript.

REFERENCES

V.B. Bajic and S.H. Seah. Dragon gene start finder: an advanced system for finding approximate locations of the start of gene transcriptional units. *Nucleic Acids Research*, 31:3560–3563, 2003.

V. B. Bajic, S. L. Tan, Y. Suzuki, and S. Sugano. Promoter prediction analysis on the whole human genome. *Nat Biotechnol*, 22(11):1467–73, November 2004.

C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. Technical report #1551, University of Wisconsin Madison, January 2006.

R. V. Davuluri, I. Grosse, and M. Q. Zhang. Computational identification of promoters and first exons in the human genome. *Nat Genet*, 29(4):412–417, December 2001.

T. A. Down and T. J. P. Hubbard. Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Res*, 12:458–461, 2002.

T. Fawcett. Roc graphs: Notes and practical considerations for data mining researchers. Technical report hpl-2003-4, HP Laboratories, Palo Alto, CA, USA, January 2003.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of the European Conference on Machine Learning*, pages 137–142, Berlin, 1998. Springer.

W. J. Kent. BLAT–the BLAST-like alignment tool. *Genome Res*, 12(4):656–664, April 2002.

G. R. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Kevin Lauerdale, and Teri E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing (PSB)*, pages 566–575. World Scientific, 2002.

V. Matys, O.V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss, P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A.E. Kel, and E. Wingender. TRANSFAC and its module TRANSCompel: Transcriptional gene regulation in eukaryotes. *Nucleic Acids Res.*, 34:D108–110, 2006.

C. E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, VIII(4), October 1978.

U. Ohler, G. C. Liao, H. Niemann, and G. M. Rubin. Computational analysis of core promoters in the drosophila genome. *Genome Biol*, 3(12):RESEARCH0087, 2002.

Kim D. Pruitt, Tatiana Tatusova, and Donna R. Maglott. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucl. Acids Res.*, 33(S1):D501–504, 2005.

G. Rätsch, S. Sonnenburg, and B. Schölkopf. RASE: Recognition of alternatively spliced exons in *C. elegans*. *Bioinformatics*, 21(Suppl. 1):i369–i377, June 2005.

S. Sonnenburg, G. Rätsch, S. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 2006. Accepted.

S. Sonnenburg, G. Rätsch, and B. Schölkopf. Large scale genomic sequence SVM classifiers. In S. Dzeroski, editor, *Proceedings of the 22nd International Conference on Machine Learning, ICML*. ACM Press, 2005.

Y. Suzuki, R. Yamashita, K. Nakai, and S. Sugano. dbTSS: Database of human transcriptional start sites and full-length cDNAs. *Nucleic Acids Res*, 30(1):328–331, January 2002.

T. Werner. The state of the art of mammalian promoter recognition. *Brief Bioinform*, 4(1):22–30, March 2003.

R. Yamashita, Y. Suzuki, H. Wakaguri, K. Tsuritani, K. Nakai, and S. Sugano. dbTSS: Database of human transcription start sites, progress report 2006. *Nucleic Acids Res*, 34:D86–89, January 2006. Database issue.

[10] They vary very slightly as for smaller chunk sizes TSS more often fall into two chunks.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e481–e488
doi:10.1093/bioinformatics/btl237

A computational approach toward label-free protein quantification using predicted peptide detectability

Haixu Tang[1,2,3], Randy J. Arnold[3,4], Pedro Alves[1], Zhiyin Xun[4], David E. Clemmer[3,4], Milos V. Novotny[3,4], James P. Reilly[3,4] and Predrag Radivojac[1,*]

[1]School of Informatics, Indiana University, Bloomington, IN, USA, [2]Center for Genomics and Bioinformatics, Indiana University, Bloomington, IN, USA, [3]National Center for Glycomics and Glycoproteomics, Indiana University, Bloomington, IN, USA and [4]Department of Chemistry, Indiana University, Bloomington, IN, USA

ABSTRACT

Summary: We propose here a new concept of peptide detectability which could be an important factor in explaining the relationship between a protein's quantity and the peptides identified from it in a high-throughput proteomics experiment. We define peptide detectability as the probability of observing a peptide in a standard sample analyzed by a standard proteomics routine and argue that it is an intrinsic property of the peptide sequence and neighboring regions in the parent protein. To test this hypothesis we first used publicly available data and data from our own synthetic samples in which quantities of model proteins were controlled. We then applied machine learning approaches to demonstrate that peptide detectability can be predicted from its sequence and the neighboring regions in the parent protein with satisfactory accuracy. The utility of this approach for protein quantification is demonstrated by peptides with higher detectability generally being identified at lower concentrations over those with lower detectability in the synthetic protein mixtures. These results establish a direct link between protein concentration and peptide detectability. We show that for each protein there exists a level of peptide detectability above which peptides are detected and below which peptides are not detected in an experiment. We call this level the minimum acceptable detectability for identified peptides (MDIP) which can be calibrated to predict protein concentration. Triplicate analysis of a biological sample showed that these MDIP values are consistent among the three data sets.

Contact: predrag@indiana.edu

1 INTRODUCTION

Rapid and reliable identification of thousands of peptides from a complex protein mixture sample using liquid chromatography tandem mass spectrometry (LC/MSMS) and other MS related technologies has established the foundation of high throughput proteomics experiments. Quantitative proteomics, i.e. quantifying proteins in a complex sample, or comparing protein abundances across different samples, however, often requires additional experimental strategies. Several labeling techniques applied to various MS instruments including isotopic coded affinity tag (ICAT) (Gygi *et al*., 1999), mass-coded abundance tagging (MCAT) (Cagney and Emili, 2002), stable isotopic labeling (Oda *et al*., 1999) and global internal standard technology (GIST) (Chakraborty and Regnier, 2002), were developed to profile the differential protein expression of two samples. In spite of their success in some quantitative proteomics experiments, these approaches have their own limitations. For example, some of them target one or several specific amino acids (e.g. ICAT targets Cys and MCAT targets Lys) and thus are limited to those proteins/peptides containing the amino acid that is modified by the reagent. A more important limitation of these approaches is that they all require performing a proper chemical reaction prior to the proteomics analysis. In addition to the expense of chemical reagents involved in this procedure, it remains unclear how the efficiency of these reactions and the protein capturing techniques used in the procedure will affect the quantification of different proteins (Zhang and Regnier, 2002).

Label-free protein quantification approaches attempt to quantify relative protein abundances directly from high-throughput proteomics analyses without applying labeling techniques. Different measures that can be derived from proteomics experiments and presumably correlated to protein abundance were proposed for different MS instruments. For instance, the integration of extracted ion chromatogram (XIC) peaks is thought to be a good measure for LC/MS experiments (Higgs *et al*., 2005) and sophisticated data analysis tools have been proposed to improve its accuracy (Leptos *et al*., 2006). In addition, it has been shown that the spectral count, i.e. the number of times a particular peptide is identified in an experiment, is correlated with the number of protein copies in the sample. Spectral counts have been successfully used to quickly estimate large changes in protein abundance (Pang *et al*., 2002; Gao *et al*., 2003), however the method appears to be significantly less sensitive when the count is relatively small and/or when the difference in protein abundance is 1–2 orders of magnitude (Liu *et al*., 2004; Bonner and Liu, 2006). In summary, there is still lack of systematic testing of the accuracy, robustness and applicability of the label-free protein quantification methods across different MS platforms.

*To whom correspondence should be addressed at School of Informatics, Indiana University, 901 East 10th Street, Bloomington, IN 47408, USA

Here we propose a new approach to label-free protein quantification in high-throughput proteomics experiments based solely on peptide identification, a method that has already been shown to be quite reliable, by learning and applying peptide features to increase the reliability and accuracy of protein quantification. It is commonly observed that the sequence coverage of identified peptides differs from one protein to another in the same proteomics experiment. One may hypothesize that the number of identified peptides or sequence coverage of a protein is highly correlated to its abundance, because the more protein copies in the sample, the higher chance a peptide derived from this protein will be identified (Washburn *et al*., 2001; Ishihama *et al*., 2005; Nesvizhskii and Aebersold, 2005). Although it is intuitively sound, it is not the case in practice. For example, in the analysis of an artificial protein mixture sample, even though twelve proteins were mixed at about the same concentration, the resulting sequence coverage of proteins based on identified tryptic peptides were very different, ranging from almost full coverage to no coverage (Purvine *et al*., 2004). This indicates that the abundance of a protein (or a tryptic peptide from it) is not the only dominant factor that determines whether or not a particular peptide can be observed in a proteomics experiment (Kuster *et al*., 2005).

Several factors related to the nature of the peptides clearly explain the fact that some peptides have higher chances of being missed in the identification than the others even though they are from the same abundant proteins in the sample. Let us use the commonly utilized platform, trypsin digestion coupled with LC/MS analysis, as an example. Peptides with masses smaller than 200 Da and greater than about 6000 Da produce ions (as +1, +2, or +3 ions) that are beyond the m/z range analyzed by the mass spectrometer, typically 200 to 2000 Da, and will not be observed. Other peptides will be so hydrophobic (water-insoluble) that they are not soluble in the LC mobile phase. Still others will be so hydrophilic (water-soluble) that they are not retained by the LC stationary phase in the sample trapping column. In both cases, the peptides will not be ionized for analysis by mass spectrometry. The amino acid composition of some peptides, such as those with multiple acidic residues, may dictate that they do not ionize efficiently in the mass spectrometer ion source. Alternatively, a peptide might ionize well but produce a fragmentation pattern in the MS/MS spectrum that cannot be easily interpreted. Some predicted peptides might never be generated because they exist in a region of the protein's structure that is very stable and thus resistant to proteolysis by trypsin. Finally, each peptide will typically co-elute from the chromatography with other peptides against which it must compete for limited ionizing protons in the electrospray ionization process.

Although these factors are relatively simple and understandable when considered separately, determining the reason for the absence of a peptide is often not straightforward. In fact, it is likely that multiple factors contribute to the overall result—lack of identification. We attempt to learn these 'factors' that govern the likelihood of identifying a peptide by a data driven approach, thus subtract them from the direct correlation between peptide identification and protein quantification, and finally obtain an accurate measure of protein abundance using peptide identification.

This paper is organized as follows. First, we introduce the notion of peptide detectability and discuss its relationship to protein quantification. Next, we show that peptide detectability can be predicted solely from the protein's primary structure with useful accuracy and analyze the sequence features most important for this process. Then, we propose a computational method to quantify a specific protein by using the coverage of identified peptides from a proteomics experiment as well as the predicted peptide detectability. Finally, we demonstrate the robustness of this approach by replicated proteomic analysis on the same sample.

2 PEPTIDE DETECTABILITY

There are four classes of factors that govern the likelihood of observing a peptide in a proteomics experiment: (i) the chemical properties of the peptide (and its parent protein); (ii) the limitation of the peptide identification protocol, including the pre-processing of the sample, the MS instruments and software tools used for mass spectrum analysis; (iii) the abundance of the peptide in the sample; and (iv) the other peptides present in the sample that compete with this peptide in the identification procedure. We define the *detectability* of a peptide as the probability that the peptide will be observed in a standard sample analyzed by a standard proteomics routine. Specifically, we are investigating data from samples treated by trypsin digestion followed by reversed-phase liquid chromatography tandem mass spectrometry in an ion trap and searched against known protein sequences using Mascot (Perkins *et al*., 1999). By standard sample we mean the sample has a fixed number of different proteins (peptides) and they are mixed at the same fixed concentration (e.g. 1 pmol/injection). We stress that, by this definition, peptide detectability is an intrinsic property of a peptide that is determined by its primary sequence as well as its location within the context of the entire protein. Peptides with higher detectabilities have a greater chance of being identified than those with lower detectabilities. As a result, if a peptide with low detectability is identified in a sample, it indicates that this peptide (or the protein this peptide is from) has a high abundance; if a peptide with high detectability is missed (not identified) in a sample, it indicates that this peptide (or the protein this peptide is from) has a low abundance. In addition, a situation in which a peptide with very low detectability is identified while those with higher detectabilities are not, suggests a false positive identification. Therefore, the notion of peptide detectability may be used to establish a direct correlation between peptide identification and protein identification/quantification.

Given a protein, we anticipate that the detectability of all tryptic peptides can be predicted from their sequences. It is, however, important to generate a sample that satisfies the standard conditions we described above, as the learning set for such a prediction. An artificial sample (sample B in Section 3) mixed from 12 model proteins in the similar concentration (1 pmol/microliter) was prepared and analyzed using LC/MS (see Section 5 for details) and the identification results were used as a learning data set for a predictor of peptide detectability in LC/MS experiments. We note that a normal (cellular) proteome sample is not completely suitable for training purposes because proteins in these types of samples have different and unknown abundances.

3 PREDICTION OF PEPTIDE DETECTABILITY

Data sets. We used four groups of data sets of mass spectra in this paper. The first group (data set A) was generated as a standard protein mixture consisting of 12 model proteins and 23 model peptides mixed at similar concentrations from 73 to 713 nM for proteins and from 50 to 1800 nM for peptides (Purvine

Table 1. Composition (fmol per one microliter injection) of six mixtures of 13 model protein chains (12 proteins). This mixture constitutes six data sets: B and B_1–B_5. See Section 5 for detailed description of the sample preparation protocols. MW indicates molecular weight

Protein	Swiss-Prot ID	MW (kDa)	B_1	B_2	B_3	B_4	B_5	B
Serum albumin, bovine	P02769	66.4	3000	300	1000	30	100	1000
Myoglobin, horse	P68082	17.0	3000	300	1000	30	100	1000
Beta-casein, bovine	P02666	23.6	1000	3000	100	300	30	1000
Catalase, bovine	P00432	59.8	1000	3000	100	300	30	1000
Lactoferrin, bovine	P24627	76.1	300	30	3000	100	1000	1000
Lysozyme, chicken	P00698	14.3	300	30	3000	100	1000	1000
Alpha-casein, bovine	P02662	23.0	100	1000	30	3000	300	1000
Pyruvate kinase, rabbit	P11974	57.9	100	1000	30	3000	300	1000
Ovalbumin, chicken	P01012	42.8	30	100	300	1000	3000	1000
DNase I, bovine	P00639	29.1	30	100	300	1000	3000	1000
RNase A, bovine	P61823	13.7	30	100	300	1000	3000	1000
Hemoglobin alpha, human	P69905	15.1	2000	2000	2000	2000	2000	2000
Hemoglobin beta, human	P68871	15.9	2000	2000	2000	2000	2000	2000

Table 2. Summary of the four data sets used in this study. Protein chains with less than 10% sequence coverage were eliminated from all data sets

Data set	Protein chains	Total tryptic peptides	Identified peptides
A	11	346	100
B	13	294	91
C	124	3403	359
D_1–D_3	200	3722	526

et al., 2004). The second group consisted of six data sets (data sets B and B_1–B_5), prepared in our labs, each representing a mixture of the same 13 model protein chains. To mimic a similar peptide competition environment in the LC/MS analysis, we intentionally mixed similar total amounts of protein in each sample as indicated in Table 1. The third group is a data set (data set C) generated from a real rat proteome, as described later. The last group consists of three data sets (data sets D_1–D_3) representing three replicate analyses of the fruit fly head proteome. With the exception of data set C, all samples were reduced and alkylated with iodoacetamide prior to trypsin digestion. The rat samples were digested in the presence of an acid-labile surfactant. All MS experiments were carried out on an ion trap mass spectrometer, either a 3-D ion trap (data sets A, C, and D) or a linear ion trap (data set B). The low m/z cut-off was between 250 and 400, and the high m/z cut-off was between 1500 and 2000 for all experiments.

Due to the large differences in protein concentrations in the whole cell lysates, we included in our analysis and learning procedures only those proteins whose coverage of identified peptides was 10% or higher. In the case of the synthetic sample by Purvine *et al*. (2004), one of the proteins contained only one identified peptide and was also removed from the subsequent analysis. The total number of protein chains, the number of tryptic peptides and the number of identified peptides in each data set are summarized in Table 2.

Machine learning methodology. Given an unseen n-residue long protein sequence $S = s_1 s_2 \cdots s_n$ and a database of peptides already detected by Mascot with high confidence, we construct a model that can approximate the probability of detecting any particular tryptic peptide from S with the same confidence. We denote this probability as $P(score(s_{[i,\ j]}) \geq t \mid S)$, where $s_{[i,\ j]} = s_i s_{i+1} \ldots s_j$ is a residue sequence of a tryptic peptide from S and t is an appropriately selected Mascot threshold (by default 40 in all our experiments). In the case when a Pro residue directly follows a basic residue (Arg or Lys) the peptide was extended until the first accessible Arg/Lys or until the C-terminus. As previously mentioned, in order to reduce the dependency of the detectability on the concentration of the protein in a cell, only proteins with $\geq$10% sequence coverage of the detected peptides were used in our analysis. All peptides whose m/z was outside of the instrument range were eliminated from training and testing as trivial.

Data representation. To enable learning, each input peptide sequence $s_{[i,\ j]}$ was represented by a fixed-length vector of real- or discrete-valued features. Two groups of features were considered: those that depend on $s_{[i,\ j]}$ only and those that also depend on the flanking regions. Thus, an identical peptide observed in the contexts of different sequence neighborhoods will in general have different detectability. The following groups of features were constructed solely from $s_{[i,\ j]}$: (i) amino acid compositions in the peptide; (ii) length of the peptide, i.e. $j - i + 1$; (iii) ion mass m($s_{[i,\ j]}$); (iv) N- and C-terminal residues, s_i and s_j; (v) sequence complexity (Wootton and Federhen, 1996); (vi) physicochemical properties averaged over the entire peptide—aromatic content and hydrophobicity (Kyte and Doolittle, 1982) and (vii) predictions obtained from various bioinformatics tools and averaged over $s_{[i,\ j]}$—namely, protein flexibility predictors (Radivojac *et al*., 2004; Vihinen *et al*., 1994), hydrophobic moment (Eisenberg *et al*., 1984), and predictions of intrinsic disorder (Obradovic *et al*., 2003; Romero *et al*., 2001; Vucetic *et al*., 2003). Since the detectability of the peptide may also be influenced by the neighboring regions, the composite features from (vii) were averaged over the regions of ±5, ±10, and ±15 residues flanking both sides of $s_{[i,\ j]}$. In addition, the residue at position s_{j+1} was also accounted for. Individual amino acids were encoded using orthogonal data representation (Qian and Sejnowski, 1988) while the compositional features were encoded by real values. Overall, the total number of features was 175. A binary class label was finally added

Table 3. Fifteen best features estimated using the t-test on data set B. Features of the same type, but averaged over flanking regions of different sizes, are presented only for the best performing window. Window ±15 indicates that the feature is averaged over $s_{[i-15,\, j+15]}$

Feature	Window	p-value	Correlation	Reference
Vihinen *et al*. flexibility	±15	$3.1 \cdot 10^{-10}$	−	Vihinen *et al*. (1994)
Hydrophobic moment	±15	$6.0 \cdot 10^{-10}$	−	Eisenberg *et al*. (1984)
B-factor prediction	±15	$2.9 \cdot 10^{-9}$	−	Radivojac *et al*. (2004)
VL2 disorder	±15	$1.3 \cdot 10^{-7}$	−	Vucetic *et al*. (2003)
Sequence complexity	0	$1.8 \cdot 10^{-7}$	+	Wootton and Federhen (1996)
VL2V disorder	±15	$3.5 \cdot 10^{-6}$	−	Vucetic *et al*. (2003)
VLXT disorder	±15	$4.1 \cdot 10^{-6}$	−	Romero *et al*. (2001)
VL2S disorder	±15	$4.3 \cdot 10^{-5}$	−	Vucetic *et al*. (2003)
VL3 disorder	±15	$5.5 \cdot 10^{-5}$	−	Obradovic *et al*. (2003)
Composition of Lys	0	$3.3 \cdot 10^{-4}$	−	N/A
Mass/length ratio	0	$1.0 \cdot 10^{-3}$	−	N/A
VL2C disorder	±15	$4.1 \cdot 10^{-3}$	−	Vucetic *et al*. (2003)
Composition of Val	0	$1.6 \cdot 10^{-2}$	+	N/A
Length	0	$1.8 \cdot 10^{-2}$	+	N/A
Composition of Gly	0	$2.1 \cdot 10^{-2}$	+	N/A

to each feature vector; 1 (positive) for a detected peptide and 0 (negative) otherwise.

Model selection. To build predictors we employed ensembles of 30 two-layer feed-forward neural networks trained using the resilient backpropagation algorithm (Riedmiller and Braun, 1993). Due to the asymmetric class sizes and small positive set (detected fragments), each network was trained on a balanced selection of positive and negative examples. Each individual training set contained all the examples from the positive class and the same number of randomly selected negative examples. The network contained 1 output neuron, while the number of hidden neurons h was varied from $h \in \{1, 2, 4\}$. All neurons contained the logistic activation function. Prior to the network training, unpromising features were eliminated using the t-test filter in which features whose p-values were above a given threshold t_{fs} were eliminated. The threshold t_{fs} for feature selection was varied from $t_{fs} \in \{0.01, 0.1, 1\}$. Note that in the case of $t_{fs} = 1$, all features were retained. Finally, correlated features were removed by employing principal component analysis and retaining 95% of the variance. A validation set containing 20% of the training data was used for model selection and overfitting prevention for each of the training sets in the ensemble. Thus, the final prediction was averaged over 30 different models and the single estimated accuracy is reported.

Performance evaluation. The performance of the predictor was evaluated within each data set (A to D) and also across various data sets. In the following, we refer to these two types of performance evaluation as cross-validation and out-of-sample estimation, respectively. In the first case we used a per protein 10-fold cross-validation. The entire set of available proteins D was first split into 10 non-overlapping sets $\{D_i \mid i = 1 \cdots 10\}$. In each step i, dataset $D - D_i$ was used for training while the prediction accuracy was estimated on the test set D_i. The final performance estimates were obtained as averages over all 10 iterations. In the out-of-sample case, we were interested in training and evaluating predictor performance on two independent experiments. In particular, a predictor was trained and optimized on one data set (say, data set A) and then applied and evaluated on all other data sets (say, data sets B, C and D). All twelve combinations were explored.

We measured sensitivity (*sn*)—the fraction of detected peptides correctly predicted, and specificity (*sp*)—the fraction of undetected peptides correctly predicted. Given *sn* and *sp*, the class-balanced accuracy can be calculated as $accuracy = (sn + sp)/2$. In this setup, a predictor always outputting the same class and a predictor outputting uniformly at random would have a balanced-sample accuracy of 50%. In addition to accuracy, we estimated the area under the ROC curve (*AUC*) using the trapezoid rule. Both accuracy and area under the curve are essentially unaffected by the asymmetry in class sizes.

Feature analysis. To gain insights into sequence and physicochemical properties governing peptide detectability, we analyzed features that best discriminate between identified and unidentified peptides. These features were selected using the standard two sample t-test on each feature independently. More precisely, a feature was split into two 1-D samples according to the class label and the hypothesis that these samples were generated according to the same probability distribution was tested. Even though the features may not come from a Gaussian distribution, the t-test is known to be robust to violations of this assumption. In Table 3 we present a ranking according to the increasing p-value of the 15 individually best features obtained on data set B. Nine of these features were based on the overall properties of the peptide including its neighborhood, while the top ranked features based solely on the peptide itself were sequence complexity, its length, the mass/length ratio and presence of Lys, Val, and Gly. Other data sets had similar ordering of the features (data not shown). As a general rule, it appears that peptides within flexible neighborhoods have lower detectability. On the other hand, presence of hydrophobic amino acids (Val, Gly) and peptide length were positively correlated with peptide detectability. Further work is needed toward deeper understanding of these properties.

Prediction accuracy. Predictor evaluation was performed in two steps. In the first step, a 10-fold cross-validation was used to estimate the prediction accuracy on each data set. In the second

Table 4. Results of learning peptide detectability using different training and testing sets. Each field contains balanced sample accuracy (*accuracy*) [%] and the area under the ROC curve (*AUC*) [%] for a particular training/test set combination

accuracy/AUC	Training set A	B	C	D_1–D_3
Test set				
A	75.8/79.7	74.8/80.3	68.0/72.0	63.0/79.2
B	68.3/77.5	65.5/70.0	62.8/69.6	62.7/68.7
C	66.7/74.6	66.8/73.5	75.0/84.0	68.0/78.1
D_1–D_3	78.7/86.5	73.1/79.0	79.9/87.6	86.8/93.0

step, performance evaluation was performed across data sets, as described above. The summary of systematic evaluations is shown in Table 4. Generally, these results strongly support our hypothesis that peptide detectability is influenced by its sequence and flanking regions from the parent protein. Interestingly, the data sets can be grouped into synthetic and whole cell, based on their out-of-sample performance. For example, best out-of-sample accuracy on data sets A and B was achieved when the training sets were B and A, respectively. Training on these synthetic data sets also achieved good performance even on data sets C and D, despite small training sizes. On the other hand, the best out-of-sample performance on data set C was achieved by training on data set D, while the best out-of-sample performance on data set D was achieved by training on C.

It can be observed from Table 4 that prediction accuracies vary between 62.7% and 86.8%, with the mean accuracy of 71.0%, while the area under the curve varied between 68.7% and 93.0%, with the mean of 78.3%. Surprisingly, training on one data set and testing on another did not generally reach similar performance when the two sets were switched. On the one hand, considering the small size of synthetic data sets, such performance could be explained by normal variation. On the other hand, the differences between data sets C and D were large and could be partially explained by the different sample densities in the feature space. In particular, it appears that data sets D_1–D_3 cover only part of the feature space covered by data set C. Thus, while training on C and testing on D_1–D_3 could produce good performance results, the opposite did not hold true. In order to verify this statement we trained a separate classification model to distinguish solely between tryptic peptides from data set C and data set D. A prediction accuracy of 57.4% indicates that there exists a difference between these two samples which can partially explain the inconsistency on the out-of-sample evaluations. In addition to the sequence biases between these two sets, there are also differences in the experimental protocol that could contribute to the discrepancy in performance, e.g. the way in which cysteines were modified in the samples was different for data set C (no modification) and D (reduced and alkylated).

4 PEPTIDE DETECTABILITY AND PROTEIN QUANTIFICATION

In the previous section, we showed that our predictor can approximate detectability of a peptide from its sequence as well as from its context in the complete protein with good prediction accuracy. In this section, we show the results of utilizing the predicted peptide detectability to measure protein abundances in the sample.

Here we analyze samples B_1–B_5 using a predictor trained on sample B in which all chains were similarly abundant. Figure 1a shows the predicted detectabilities of all tryptic peptides from each protein from sample B_1. Peptides from the same protein are shown in the same column, sorted by their detectabilities. Proteins were sorted by their relative abundances (concentrations) in the mixture. The identified peptides are shown as empty squares, while the missed peptides are shown as dashes. It is clear that, for each protein in sample B_1, the identified peptides tend to have higher detectabilities than those not identified. This is consistent to the prediction accuracy results as shown in the last section. For each protein, we can determine its *minimum acceptable detectability of identified peptides* (MDIP), a cutoff value of detectability which maximizes the sum of true positive and true negative rates. If all peptides from a protein are detected, the MDIP of this protein is set to 0, and if none of the peptides from a protein is detected, the MDIP of this protein is set to 1. It can be observed from Figure 1a that the MDIP values, shown as black squares, increase as the protein abundance decreases. This trend is approximated by a solid regression line. Similar results were obtained in the remaining samples B_2–B_5 (data not shown).

We computed the MDIP for each protein in five different synthetic mixtures (B_1–B_5) and show them in Figure 1b. Each column in Figure 1b corresponds to a particular concentration and represents proteins from different experiments. For example, in column 2 the grey diamond and circle represent proteins ALBU_BOVIN and KPYM_RABIT, respectively, both with concentration 1000 fmol. However, ALBU_BOVIN was mixed at this concentration in sample B_3, while KPYM_RABIT was mixed at concentration 1000 fmol in sample B_2 (see Table 1). Similarly to the trend observed in Figure 1a, we can see from Figure 1b a linear relationship between MDIP and protein concentration. Moreover, their relationships are generally similar from one protein to the next.

Figure 2 shows the MDIP for hemoglobin A and hemoglobin B, which were mixed in the same amount in all experiments (Table 1), across different samples. It shows low variation of MDIP, suggesting it is a robust measure of protein abundance.

In the last experiment, we show that MDIP may be used as a measure of protein quantification in high throughput proteomics experiments. Here, we used three replicate data sets (D_1–D_3) to demonstrate the robustness of the protein quantification method we propose. Using the same predictor trained on data set B, we predicted the detectability of all proteins in *D. melanogaster* proteome. In each of the three experiments (D_1–D_3), we computed the MDIP score for each protein. Figure 3 shows the scatter plots of pairwise comparisons of MDIP scores between any two experiments.

5 MASS SPECTRUM ACQUISITION AND ANALYSIS

Data sets B and B_1–B_5. Mixtures of twelve standard proteins (listed in Table 1) were paired or triply-grouped such that the combined molecular weights in each group totaled about 80 to 90 kDa. Samples of each protein were prepared as stock solutions of 60, 20, and 2 micromolar concentration, or 90, 30, and 3 micromolar for

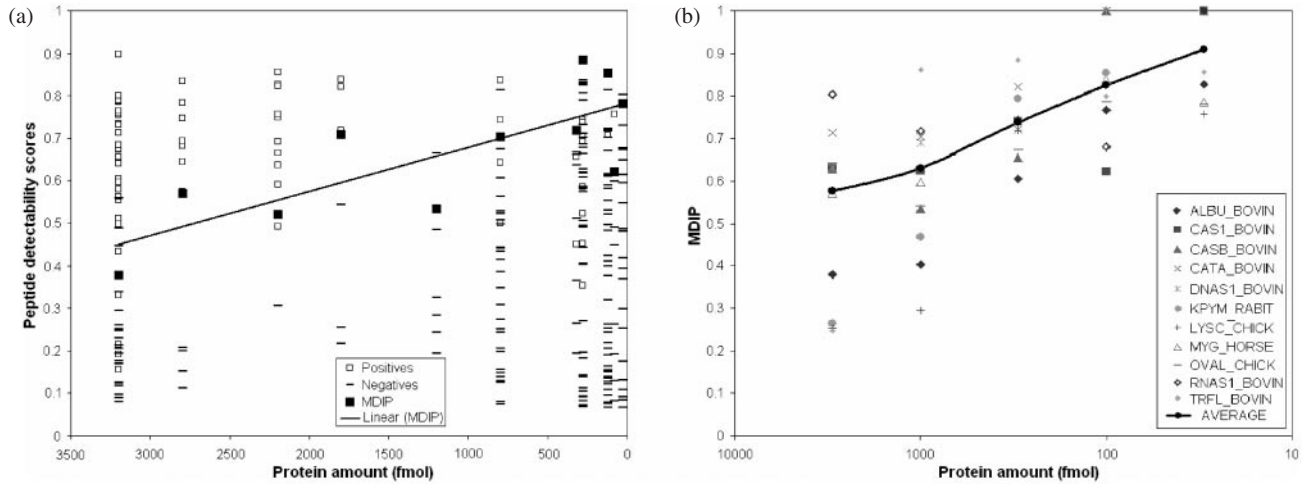


Fig 1. (**a**) Peptide detectability of proteins in sample B_1. Each column displays peptide detectabilities from the same protein. Proteins are sorted according to the decreasing concentration (from left to right), however in order to avoid overlaps, proteins with the same concentration were separated (e.g. columns 1 and 2 correspond to the amount of 3000 fmol). Peptides identified by Mascot are shown as empty squares; peptides not identified are shown as dashes. Minimum acceptable detectability of identified peptides (MDIP) is shown as black squares for each protein. (**b**) MDIP of the proteins from samples B_1–B_5 as a function of protein amount. The columns represent protein amounts and not different experiments. For example, in column 1 RNAS1_BOVIN (top detectability) corresponds to experiment B_5, while CATA_BOVIN (second highest detectability) corresponds to experiment B_2 (see Table1). Both proteins have the abundance of 3000 fmol.

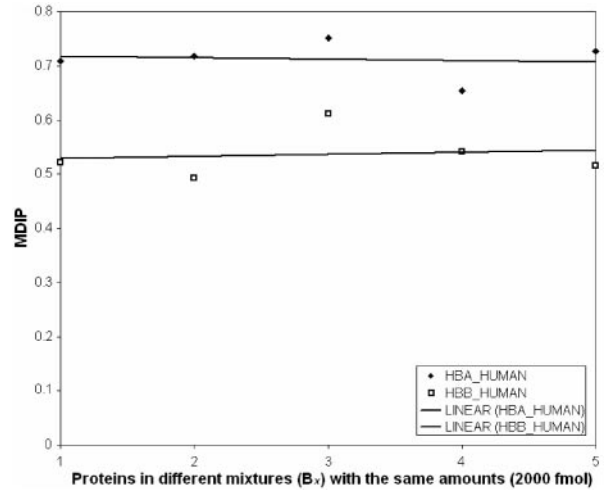


Fig 2. Minimum acceptable detectability of identified peptides (MDIP) of hemoglobin A (HBA_HUMAN, black diamonds) and hemoglobin B (HBB_HUMAN, white squares) in samples B_1–B_5. Each column x in the figure corresponds to a data set B_x.

the triply-grouped samples. Proteins were then mixed in various ratios such that the same molecular weight equivalent was present at 3000, 1000, 300, 100, and 30 fmol per microliter of final digestion solution, combined with buffer, reduced with dithiothreitol (DTT), alkylated with iodoacetamide (IAM), and digested at 37°C for 18 hours. After acidification, samples were loaded onto a 15 mm by 100 micron i.d. trapping column packed with 5-micron BioBasic 18 particles with 300 Å pores (Thermo Hypersil-Keystone, San Jose, CA). Peptides were separated using a 30-minute reversed-phase liquid chromatography gradient from 3% to 40% acetonitrile at 250 nL/min (Eksigent Technologies, Livermore, CA) on a 12 to 15 cm, 75 micron i.d. capillary column pulled to a small (~10 micron) tip and packed in-house with 5 micron C-18 coated particles (Betasil C18, Thermo Hypersil-Keystone, San Jose, CA). As peptides eluted from the column, they were electrosprayed into the source of a Thermo Electron (San Jose, CA) LTQ linear ion trap mass spectrometer and analyzed by mass spectrometry and tandem mass spectrometry. By using dynamic exclusion, the mass spectrometer was limited to acquiring only one tandem mass spectrum for a given parent m/z over a 30-second window.

Data set C. Rat brain regions (amygdala, caudate putamen, frontal cortex, hippocampus, hypothalamus, and nucleus accumbens) were digested separately with proteomics grade (modified) trypsin in the presence of an acid-labile surfactant. Tryptic peptides were separated by nano-flow reversed-phase liquid chromatography and electrosprayed directly into a ThermoFinnigan (San Jose, CA) LCQ Deca XP ion-trap mass spectrometer which recorded mass spectra and data-dependent tandem mass spectra of the peptide ions. Dynamic exclusion was employed to limit acquisition of tandem mass spectra for the same parent m/z over a 60-second window.

Data set D. Drosophila genotype: elav-GAL4 (Stock number: Bloomington/458) flies were harvested and separated according to sex at day 1 of adult life. Flies were cultured on standard cornmeal medium and maintained at 25°C. Flies ($n = 250$) were anesthetized with CO_2, flash frozen and decapitated with shaking in liquid N_2. Heads were collected on dry ice and stored at −80°C. Proteins were extracted using a mortar and pestle in 0.2 M phosphate buffer saline plus 8 M urea plus 0.1 mM phenylmethylsulfonyl fluoride (pH 7.0) solution. Proteins were centrifuged (15700 g at 4°C) for 10 minutes and the supernatant was kept for the determination of protein concentration using Bradford assay. Extracted proteins were reduced with DTT, alkylated with IAM, and digested with TPCK-treated trypsin after diluting the urea to 2 M with

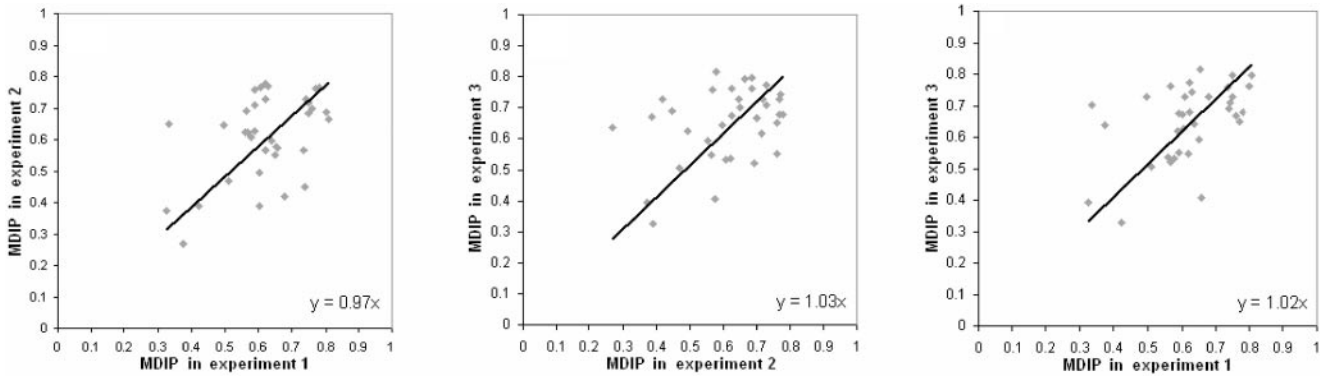


Fig 3. Scatter plots of pairwise comparisons of MDIPs of the identified proteins in samples D_1–D_3. Each dot represents a single protein identified in both experiments.

0.2 M Tris buffer (pH 8.0). Tryptic peptides were isolated by C-18 solid-phase extraction, vacuumed to dryness, and stored at −80°C until future use. Peptides from each SCX fraction were separated by nano-flow reversed-phase liquid chromatography (15 cm × 75 μm i.d. fused silica capillary column pulled to a fine tip and packed with 5 μm, 100 Å amino-terminated C-18 packing material (Michrom Bioresources, Auburn, CA), eluted with a gradient from 5 to 45% acetonitrile at 250 nL/min). Eluting peptides were electrosprayed directly into the source of a Thermo Finnigan LCQ Deca XP ion trap mass spectrometer and analyzed by MS (m/z 250–1500) and data-dependent MS/MS on the three most intense ions.

Tandem mass spectra were searched against protein sequences for the twelve known proteins (data set B), *R. norvegicus* in the Swiss-Prot database (data set C) or *D. melanogaster* (data set D) using a licensed copy of Mascot (Perkins *et al*., 1999) for peptide identification. Searches were performed with fixed modification of carbamidomethyl cysteine (where appropriate) and variable modifications of protein N-terminal acetylation and methionine oxidation selected and a maximum of one missed cleavage site. Mascot result files were parsed using a Protein Results Parser program written in-house to create training sets including all peptides with Mascot scores of 40 or higher for doubly-charged precursors. Peptides with Mascot scores below 40 were treated as negatives in the training sets.

6 CONCLUSIONS

In this study we propose a new concept of peptide detectability, an intrinsic property of a peptide in the context of its parent protein. This detectability can be used to quantify proteins from the peptide identification results in a standard proteomics experiment. We suggest that peptide detectability can be successfully approximated from its amino acid sequence and neighboring regions of its parent protein. To this goal, we carried out a controlled proteomics experiment in which all protein concentrations were similar to create a ''standard'' data set from which peptide detectability can be learned. In addition to the standard data set B we used other samples to train and evaluate neural-network predictors. Despite small and noisy data sets, these predictors achieved useful cross-validation and out-of-sample accuracies, ranging from 62% to 87%, while the areas under the ROC curves ranged from 69% to 93%.

At this stage, our work is a proof-of-concept study of utilizing the predicted peptide detectability to measure protein abundances in high-throughput proteomics experiments. Further experiments will be necessary in order to precisely determine its sensitivity. It should also be noted that, while demonstrated here as a method to improve quantitative measurements of proteins in proteomics experiments, this approach also offers promise to improve protein identification in cases where only a limited number of peptides are identified.

From the machine learning perspective, we provide only first indications that peptide detectability is predictable from the sequence of its parent protein, thus leaving substantial room for improvement. It is likely that increased data set sizes and variability of samples will contribute to the overall increase in accuracy of detectability prediction, thus somewhat compensating for the class-label noise in the real proteomic samples used in this study. This noise was in part introduced by our simplifying the original problem in which all peptides with Mascot scores <40 were labeled as negative. In addition, we believe that further improvements can be achieved by controlled proteomics experiments in which the informatics approaches proposed here could be properly calibrated.

The results presented here are based on data from a common proteomics analytical platform; nanoflow reversed-phase liquid chromatography coupled by electrospray ionization to tandem mass spectrometry in an ion trap mass spectrometer. Several other analytical methods, such as 2-D liquid chromatography, capillary electrophoresis, MALDI ionization, electron-capture/electron-transfer dissociation, and photoinduced dissociation, as well as alternative proteases are also commonly used in the analysis of complex proteomics samples. Measurements of peptide detectability for analytical platforms based on combinations of these techniques allows for further training, and the potential to determine the most sensitive analytical platform to be used for detection of a specific protein.

ACKNOWLEDGEMENTS

The authors would like to acknowledge William McBride and Wendy Strother-Robinson for help in procuring the rat brain samples, Thomas Kaufman for help in generating the drosophila samples and Narmada Jayasankar for help in data analysis. Dr. Kolker is thanked for providing us with the data generated by Purvine *et al*. (2004). This work is supported in part by the Indiana University Office of the Vice President for Research through a Faculty

Research Support grant awarded to RJA, HT and PR. MVN thanks the Indiana Genomics Initiative (INGEN) of the Indiana University, supported in part by the Lilly Endowment Inc. JPR acknowledges support from the National Science Foundation.

REFERENCES

Bonner, A.J. and Liu, H. (2006) Towards the prediction of protein abundance from tandem mass spectrometry data. *Proceedings of the SIAM International Conference on Data Mining*, **6**, 599–603.

Cagney, G. and Emili, A. (2002) De novo peptide sequencing and quantitative profiling of complex protein mixtures using mass-coded abundance tagging. *Nat. Biotechnol.*, **20**, 163–170.

Chakraborty, A. and Regnier, F. (2002) Global internal standard technology for comparative proteomics. *J. Chromatogr. A*, **949**, 173–184.

Eisenberg, D., Weiss, R.M. and Terwilliger, T.C. (1984) The hydrophobic moment detects periodicity in protein hydrophobicity. *Proc. Natl. Acad. Sci. USA*, **81**, 140–144.

Gao, J., Opiteck, G.J., Friedrichs,M., Dongre, A.R. and Hefta, S.A. (2003) Changes in the protein expression of yeast as a function of carbon source. *J. Proteome Res.*, **2**, 643–649.

Gygi, S.P., Rist, B., Gerber, S.A., Turecek, F., Gelb, M.H. and Aebersold, R. (1999) Quantitative analysis of protein mixtures using isotope coded affinity tag. *Nat. Biotechnol.*, **17**, 994–999.

Higgs, R.E., Knierman, M.D., Gelfanova, V., Butler, J.P. and Hale, J.E. (2005) Comprehensive label-free method for the relative quantification of proteins from biological samples. *J. Proteome Res.*, **4**, 1442–1450.

Ishihama, Y., Oda, Y., Tabata, T., Sato, T., Nagasu, T., Rappsilber, J. and Mann, M. (2005) Exponentially modified protein abundance index (emPAI) for estimateion of absolute protein amount in proteomics by the number of sequenced peptides per protein. *Mol. Cell Proteomics*, **4**, 1265–1272.

Kuster, B., Schirle, M., Mallick, P. and Aebersold, R. (2005) Scoring proteomes with proteotypic peptide probes. *Nat. Rev. Mol. Cell Biol.*, **6**, 577–583.

Kyte, J. and Doolittle, R.F. (1982) A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, **157**, 105–132.

Leptos, K.C., Sarracino, D.A., Jaffe, J.D., Krastins, B. and Church, G.M. (2006) MapQuant: open-source software for large-scale protein quantification. *Proteomics*, **157**, 1770–1782.

Liu, H., Sadygov, R.G. and Yates, J.R.,3rd (2004) A model for random sampling and estimation of relative protein abundance in shotgun proteomics. *Anal. Chem.*, **76**, 4193–4201.

Nesvizhskii, A.I. and Aebersold, R. (2005) Interpretation of shotgun proteomic data: the protein inference problem. *Mol. Cell Proteomics*, **4**, 1419–1440.

Obradovic, Z., Peng, K., Vucetic, S., Radivojac, P., Brown, C.J. and Dunker, A.K. (2003) Predicting intrinsic disorder from amino acid sequence. *Proteins*, **53**, 566–572.

Oda,Y., Huang, K., Cross, F.R., Cowburn, D. and Chait, B.T. (1999) Accurate quantitation of protein expression and site-specific phosphorylation. *Proc. Natl. Acad. Sci. USA*, **96**, 6591–6596.

Pang, J.X., Ginanni, N., Dongre, A.R., Hefta, S.A. and Opitek, G.J. (2002) Biomarker discovery in urine by proteomics. *J. Proteome Res.*, **1**, 161–169.

Perkins, D.N., Pappin, D.J., Creasy, D.M. and Cottrell, J.S. (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, **20**, 3551–3567.

Purvine, S., Picone, A.F. and Kolker, E. (2004) Standard mixtures for proteome studies. *OMICS*, **8**, 79–92.

Qian, N. and Sejnowski, T.J. (1988) Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.*, **202**, 865–884.

Radivojac, P., Obradovic, Z., Smith, D.K., Zhu, G., Vucetic, S., Brown, C.J., Lawson, J.D. and Dunker, A.K. (2004) Protein flexibility and intrinsic disorder. *Protein Sci.*, **13**, 71–80.

Riedmiller, M. and Braun, H. (1993) A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *Proceedings of the IEEE Interntional Conference on Neural Networks*, **1**, 586–591.

Romero, P., Obradovic, Z., Li, X., Garner, E.C., Brown, C.J. and Dunker, A.K. (2001) Sequence complexity of disordered protein. *Proteins*, **42**, 38–48.

Vihinen, M., Torkkila, E. and Riikonen, P. (1994) Accuracy of protein flexibility predictions. *Proteins*, **19**, 141–149.

Vucetic, S., Brown, C.J., Dunker, A.K. and Obradovic, Z. (2003) Flavors of protein disorder. *Proteins*, **52**, 573–584.

Washburn, M.P., Wolters, D. and Yates, J.R.,3rd (2001) Large-scale analysis of the yeast proteome by multidimensional protein identification strategy. *Nat. Biotechnol.*, **19**, 242–247.

Wootton, J.C. and Federhen, S. (1996) Analysis of compositionally biased regions in sequence databases. *Methods Enzymol.*, **266**, 554–571.

Zhang, R. and Regnier, F.J. (2002) Minimizing resolution of isotopically coded peptides in comparative proteomics. *J. Proteome Res.*, **1**, 139–147.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e489–e496
doi:10.1093/bioinformatics/btl234

An integrative approach for causal gene identification and gene regulatory pathway inference

Zhidong Tu, Li Wang, Michelle N. Arbeitman, Ting Chen and Fengzhu Sun*

Molecular and Computational Biology Program, University of Southern California, Los Angeles, USA

ABSTRACT

Motivation: Gene expression variation can often be linked to certain chromosomal regions and are tightly associated with phenotypic variation such as disease conditions. Inferring the causal genes for the expression variation is of great importance but rather challenging as the linked region generally contains multiple genes. Even when a single candidate gene is proposed, the underlying biological mechanism by which the regulation is enforced remains unknown. Novel approaches are needed to both infer the causal genes and generate hypothesis on the underlying regulatory mechanisms.

Results: We propose a new approach which aims at achieving the above objectives by integrating genotype information, gene expression, protein-protein interaction, protein phosphorylation, and transcription factor (TF)–DNA binding information. A network based stochastic algorithm is designed to infer the causal genes and identify the underlying regulatory pathways. We first quantitatively verified our method by a test using data generated by yeast knock-out experiments. Over 40% of inferred causal genes are correct, which is significantly better than 10% by random guess. We then applied our method to a recent genome-wide expression variation study in yeast. We show that our method can correctly identify the causal genes and effectively output experimentally verified pathways. New potential gene regulatory pathways are generated and presented as a global network.

Availability: Source code is available upon request.

Contacts: fsun@usc.edu

1 INTRODUCTION

Gene expression variation has been observed in human, yeast and other organisms (Brem, *et al*., 2002; Morley, *et al*., 2004; Turk, *et al*., 2004). By linkage analysis, the variation of gene expression can often be explained by the variation of DNA sequences on chromosomes. Great interests have been arisen in finding the causal genes and the mechanisms which account for the expression variation (Friedman, *et al*., 2000; Brem, *et al*., 2002; Yvert, *et al*., 2003; Bing and Hoeschele, 2005; Li, *et al*., 2005; Schadt, *et al*., 2005; Li, *et al*., 2006).

In these studies, the expression level is treated as a quantitative trait and the genetic loci linked to the trait are usually termed as eQTL (expression quantitative trait loci). Determining the eQTL, however, doesn't answer which genes are the causal genes for the expression variation since tens or even more genes can be contained in the eQTL. Although further fine mapping will reduce the confidence interval of the eQTL, it is both time consuming and laborious. Other factors, such as high linkage disequilibrium could make fine mapping less powerful. Even when the number of candidate genes is reduced to be manageable, the underlying mechanism by which the regulation is enforced remains unknown.

Inferring the casual genes is challenging but very important in disease studies (Schadt, *et al*., 2005). Simple method based on expression correlation was proposed but without solid verification and large amount of potentially useful information were ignored (Bing and Hoeschele, 2005). Gene expression regulation is traditionally divided into *cis*-regulation and *trans*-regulation. If the eQTL are close to the target gene itself (*cis*-regulation), then the DNA variation is most likely happened in the transcription regulatory regions of the gene, such as the promoter, enhancer, etc. In *trans*-regulation, eQTL are far away from the target gene. In this case, the causal genes can be transcription factors (TFs) which regulates the target gene or genes which affect the activity of the TFs. We are particularly interested in *trans*-regulation as *cis*-regulation is relatively trivial to identify. For eQTL containing a TF which bind to the promoter region of the target gene, the TF is a good candidate for the causal gene. However, in many cases, the eQTL do not contain any TFs (Brem, *et al*., 2002; Yvert, *et al*., 2003). An alternative factor therefore must exist and need to be identified. One possible mechanism by which this alternative factor regulates the target gene expression is by regulating the TFs. Through protein-protein interaction and other mechanisms such as phosphorylation, signal is conveyed from this alternative factor to the TF and eventually alters the expression of the target gene. Such (signaling) pathways have been widely found in multiple biological processes and are considered to be one of the most fundamental gene expression regulatory mechanisms in biological systems (Ogawa, *et al*., 2000; Yoshimoto, *et al*., 2002; Dodge-Kafka, *et al*., 2005).

''Pathway'' is frequently referred in recent publications but with rather different meanings (Steffen, *et al*., 2002; Tian, *et al*., 2005). We define a pathway as a set of both directionally and undirectionally connected proteins which contains at least one TF at one end. (We don't distinguish gene and its protein product and they are used interchangeably throughout the manuscript.) Both the proteins involved and the topologies are considered as pathway components. Slightly different from (Steffen, *et al*.,

*To whom correspondence should be addressed.

2002), we don't require the pathway to be strictly linear (i.e., we allow network structures) to make more realistic modeling. A conceptual pathway is shown in Figure 1.

Although related to the research on ''transcription regulatory network'' which aims at inferring the regulatory relationship among transcription factors (Friedman, *et al*., 2000; Basso, *et al*., 2005; Rogers and Girolami, 2005; Xing and van der Laan, 2005), our work is quite different. On the surface, the difference appears as we consider the whole gene network (protein-protein interaction, protein phosphorylation, and TF-DNA binding) instead of only TFs. Down to the detail, our interests are not in identifying neighbor-to-neighbor regulations. Instead, we are identifying pathways linking causal genes and target genes to explain the regulatory relationships between them. In Figure 1, a link between gene A and gene B doesn't indicate that A regulates B's expression, which is usually the case for transcription regulatory network inference. Here, it stands for that protein A affects the expression of the target gene by interacting with protein B. Despite of these differences between the pathway and ''transcription regulatory network'', connections do exist as the transcription regulation could be part of the pathway (or even the full pathway in some cases). We'll give more details of the differences and connections between our method and previous approaches in the Methods section.

Several approaches have been proposed to systematically identify function modules, pathways and motifs in the biological system (Ideker, *et al*., 2002; Yeger-Lotem, *et al*., 2004; Qi, *et al*., 2005; Pan, *et al*., 2006). Algorithms are designed specifically for pathway identification (Steffen, *et al*., 2002; Scott, *et al*., 2005). Although these algorithms can successfully find known pathways, huge numbers of other ''pathways'' are also generated. The high false positive rate significantly limits its application to solving real biological problems. Another approach proposed by Yeang *et al*. is very successful when it is applied to a manually selected subnetwork. However, as the algorithm requires large amount of perturbation data, it's much less competent when applied to genome-wide analysis (Yeang, *et al*., 2004; Yeang, *et al*., 2005). Rather than finding all the ''possible'' pathways, we try to locate the functioning ones which can be revealed by analyzing experimental data.

We first designed a test to verify our method quantitatively. Rosetta compendium data set (Hughes, *et al*., 2000) was used for this purpose which interrogated expression profiles of 276 deletion mutants. We show that over 40% of the inferred causal genes are correct, which is more than 4 times better compared with 10% by random guess. We then applied our method to a recent genome wide expression variation study in yeast (Brem, *et al*., 2005). We demonstrate that experimentally verified causal genes and pathways can be correctly inferred and we also propose new potential pathways.

2 METHODS

An overview of our multi-step procedure is shown in Figure 2. For a target gene, the procedure identifies the eQTL by linkage analysis using expression profile and genotype information. This generates a list of genes which contains the real causal gene for the target gene expression variation. Gene network is compiled by integrating protein-protein physical interactions, protein phosphorylations and TF-DNA binding information. As the kernel of the whole process, we designed a network based stochastic inference algorithm to identify the most likely causal genes in the eQTL and the underlying pathway.

2.1 Basic assumptions

Given the target gene, the list of candidate causal genes, gene expression profiles and the network, we infer the most likely causal genes and the underlying pathways. Two assumptions are made. First, since our focus is on *trans*-regulation, we assume that the causal gene regulates the target gene by affecting the activities of the TF(s) for the target gene through a pathway. This assumption holds for most known cellular pathways. Although other regulatory mechanisms do exist, we don't explicitly consider them in this study. Second, we assume that the activities of genes on the pathway correlate with target gene's expression. The idea is illustrated by Figure 1(b). When a gene on the pathway is inactivated (e.g., knocked out), the expression of the target gene is either down-regulated if the inactivated gene has a positive effect or up-regulated otherwise. Since the activity of gene product is hard to measure directly, we use gene's expression level to approximate it. Clearly, this approximation could be violated as protein activity is also regulated by post-translational regulations such as phosphorylation. However, such approximation is still widely in use and certain successes have been reported (Segal, *et al*., 2003). We will revisit this issue in the discussion section. Zien *et al*. found that genes on the same pathway had higher ''synchrony'' in their expressions and this supports our second assumption (Zien, *et al*., 2000).

2.2 Searching the network

Based on the assumptions described above, the problem can be rephrased as to find the pathway which starts from the causal gene and ends at the TFs regulating the target gene so that the expression of the genes on the pathway are correlated with the target gene. We designed a network based stochastic backward searching algorithm to solve the problem. The stochastic model is chosen over deterministic ones mainly due to two reasons. First, the biological system itself can be modeled as a stochastic network with various interactions occurring with different probabilities. It's natural to design an algorithm which acknowledges the uncertainties in the system. Second, deterministic algorithms require the pathway length to be determined in advance and the length cannot be too large due to high computation complexity. They also require the pathway be strictly linear (Steffen, *et al*., 2002; Scott, *et al*., 2005). All these issues can be avoided with a stochastic algorithm.

The basic idea of our algorithm can be described as follows. We start from a TF and initiate a ''walk'' by following edges in the network. Decisions on what edges to take depend on gene expression profile in a non-deterministic fashion. Genes in eQTL will be visited at different frequencies. The genes with higher frequencies are more likely to be the causal genes and the most frequently traveled paths are regarded as the underlying regulatory pathways. We formalize the algorithm as follows.

For a target gene g_t, the set of transcription factors binding to it are denoted as $T_{g_t} = (t_1, \ldots, t_n)$, the candidate causal genes in the eQTL regions are denoted as $C_{g_t} = (g_{c_1}, \ldots, g_{c_m})$. The gene network is represented as a graph $\mathbf{G}$ in which the protein-protein interactions are represented as undirected edges while protein phosphorylation and TF-DNA bindings are represented as directed edges. For each $t_k \in T_{g_t}$, we start a stochastic search procedure as shown in Figure 3.

We denote all the neighbors of a particular gene in the gene network as $Nei(\cdot)$, so that $b \in Nei(a) \Leftrightarrow e_{ba} \in \mathbf{G}$, where e_{ba} represents a directed edge from b to a. Starting from t_k, we estimate for each $g_i \in Nei(t_k)$ the likelihood that g_i is the cause for the expression variation of the target gene g_t. Based on our second assumption, we estimate such causal effect by the absolute value of the Pearson correlation coefficient of the expressions between g_i and g_t, denoted as $|\rho_{(g_i, g_t)}|$. Intuitively, a gene showing strong expression correlation with the target gene has a higher probability of being involved in the pathway. However, as not all the genes on the pathway necessarily correlate with the target gene due to other post-translational regulation mechanisms, we give non-correlated genes a residual chance for being on the regulatory pathway by defining the casual effect of g_i with respect to g_t as $\xi(g_i, g_t) = \max\{|\rho(g_i, g_t)|, \varepsilon\}$, where $0 < \varepsilon < 1$ is the residual causal effect a non-correlated gene could have upon g_t.

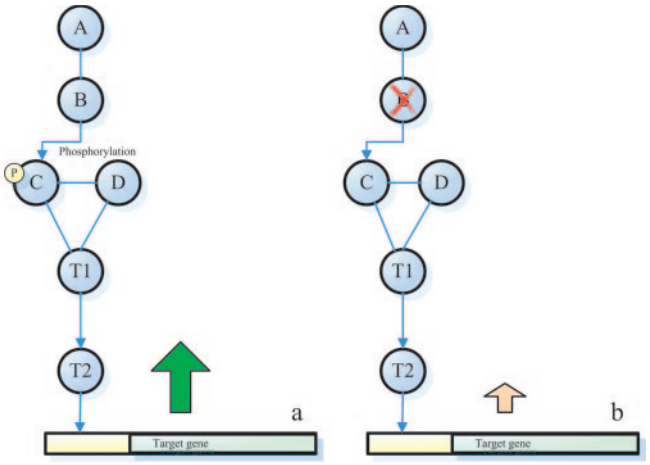


Fig. 1. A conceptual gene regulatory pathway. (a). Genes involved in the pathway are shown as circles (A,B,C,D,T1 and T2). B represents a kinase which activates downstream protein C by phosphorylation. T1 and T2 are transcription factors and T1 positively regulates T2's expression. T2 binds to the promoter region of the target gene and activates its expression. Edges without arrow indicate protein-protein interactions and edges with arrow imply the transcriptional regulations or phosphorylations. (b) Gene B on the pathway is inactivated. The expression of the target gene is down-regulated as consequence. We don't require pathway be strictly linear so that indispensable components of the pathway (e.g., D) can be included.

We denote a path as $P(g_0, g_1, \ldots, g_z)$, where $g_0, g_1, \ldots, g_z$ are nodes in the graph and cycles are not allowed in the path, i.e., $g_i \neq g_j$ for any g_i, g_j on the path. To ensure paths are non-cyclic, a set $\mathbf{U}$ is introduced which contains only unvisited genes. We stochastically select $g_i \in Nei(t_k) \cap \mathbf{U}$ and transit from t_k to g_i. The transition probability is determined by equation (1). Based on this transition probability, unvisited neighbor genes with greater causal effect will have higher chances of being visited next. The chosen gene will be removed from $\mathbf{U}$ thereafter.

$$\Pr\{g_i \mid t_k, g_i \in Nei(t_k) \cap \mathbf{U}\} = \frac{\xi(g_i, g_t)}{\sum_{g_s \in Nei(t_k) \cap U} \xi(g_s, g_t)}. \quad (1)$$

After we arrive at g_i, the same procedure is repeated. We select $g_i' \in Nei(g_i) \cap \mathbf{U}$ based on similar transition probability as described by equation (2).

$$\Pr\{g_i' \mid g_i, g_i' \in Nei(g_i) \cap \mathbf{U}\} = \frac{\xi(g_i', g_t)}{\sum_{g_s \in Nei(g_i) \cap \mathbf{U}} \xi(g_s, g_t)}. \quad (2)$$

By noticing that we always calculate the causal effect of a gene g_i with respect to g_t, it's clear that our procedure is different from most transcription regulatory network inference algorithm. In our procedure, the objective is not to identify the relationship between connected genes (i.e., g_i and g_i'), but to find connected genes which are likely to be the cause for the expression variation of the target gene g_t.

The above procedure stops when it reaches any gene $g_i \in C_{g_t}$ or when it enters a dead end (i.e., $Nei(g_i) \cap \mathbf{U} = \emptyset$). We also set an upper bound for the total transitions allowed to ensure a stop. The upper bound is chosen to be unrealistically high for any known pathway and is different from the path length in those deterministic pathway finding algorithms. Suppose we stop at $g_c \in C_{g_t}$ after one round of the procedure, the path can be written as $P(t_k, \ldots, g_i, \ldots, g_c)$. The causal effect of g_c on g_t through $P(t_k, \ldots, g_i, \ldots, g_c)$ can be calculated by (3). Here, we assume that the causal effect of each node on the pathway is independent with each other. This assumption may not hold in reality. However, considering interactions among genes on the pathway will make the problem too complex and we do not consider them in this study.

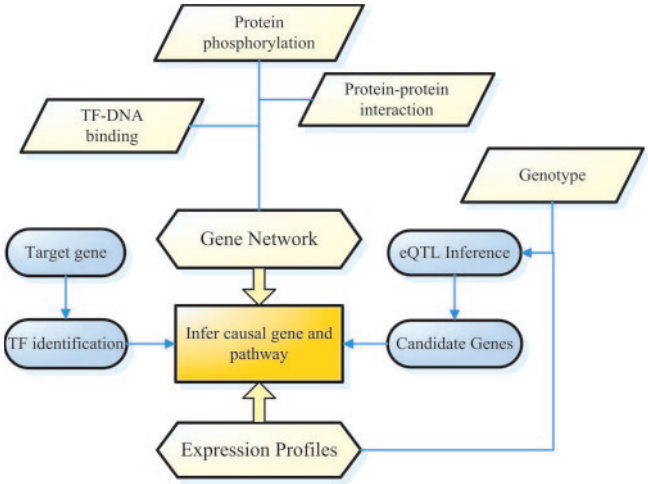


Fig. 2. Overview of our multi-step procedure for causal gene identification and gene regulatory pathway inference.

$$p(g_c, t_k, P(t_k, \ldots, g_c)) = \xi(t_k, g_t) \cdot \ldots \cdot \xi(g_c, g_t). \quad (3)$$

As equation (3) measures the causal effect of g_c on g_t with respect to a specific potential pathway, the general causal effect of g_c considering the whole gene network can be estimated by equation (4), where $P_{t_k}^{g_c}$ denotes all the paths starting from t_k and ending at g_c.

$$p(g_c, t_k) = \sum_{P_{t_k}^{g_c}} p(g_c, t_k, P(t_k, \ldots, g_i, \ldots, g_c)). \quad (4)$$

To calculate $p(g_c, t_k)$, each gene $g_i \in \mathbf{G}$ is associated with a counter $V_{t_k}(g_i)$ to record the times it's been visited. We iterate the whole procedure N times and N is set to be large enough so that (5) can be approximated, where $V_{t_k}(g_c)$ denotes the visit times for $g_c \in C_{g_t}$.

$$\lim_{N \to +\infty} V_{t_k}(g_c)/N = p(g_c, t_k). \quad (5)$$

If the target gene has more than one TF, we assign each TF a weight based on their causal effect on the target gene and linearly combine them as shown by (6). The probability that g_c is the casual gene in the eQTL considering all the TFs for the target gene g_t is estimated by (7).

$$V_T(g_c) = \frac{\sum_{k=1}^{m} \xi(t_k, g_t) V_{t_k}(g_c)}{\sum_{k=1}^{m} \xi(t_k, g_t)}. \quad (6)$$

$$\widehat{\Pr(g_c)} = \frac{V_T(g_c)}{\sum_{g_s \in C(g_t)} V_T(g_s)} = \frac{\sum_k p(g_c, t_k^c)}{\sum_{s: g_s \in C(g_t)} \sum_k p(g_s, t_k^s)}. \quad (7)$$

Since we assume there's only one causal gene in each eQTL, the gene with the largest posterior probability is reported as the cause as shown by (8).

$$g_c^* = \arg\max_{g_s \in C_{g_t}} \widehat{\Pr(g_s)}. \quad (8)$$

To identify the underlying pathway, we start from g_c^* and trace backwards. We find from $Nei(g_c^*)$ the gene with the largest visit count and move to that gene (not stochastically). We repeat until we arrive at t_k. By this way, we find the most probable pathway which links g_c^* and t_k. The linear pathway generated by this approach is mainly for simplicity consideration. As indicated by (4), there could be multiple paths connecting g_c^* and t_k, and all of them contribute to the causal effect of g_c^*.

2.3 Select subset of conditions

It's well known that TFs only actively regulate their target genes under specific conditions (Ihmels, *et al*., 2002; Harbison, *et al*., 2004; Segal, *et al*., 2004). It will therefore be beneficial to infer the pathway under these conditions to exclude the noise introduced by non-relevant conditions. To achieve this goal, we implemented two different methods.

First, we follow the signature algorithm developed by Ihmels *et al*. to select appropriate subset of conditions (Ihmels, *et al*., 2002). Suppose the expression levels of gene g_t are measured under M conditions in the original data set, denoted as $O^1_{g_t},\ldots,O^M_{g_t}$. A condition m is selected if it satisfies equation (9), where $\overline{O_{g_t}}$ is the average expression level and σ_{g_t} is the standard deviation. We empirically choose τ equal to 1 to ensure both sufficient variation and enough number of included conditions.

$$\frac{|O^m_{g_t} - \overline{O_{g_t}}|}{\sigma_{g_t}} > \tau. \tag{9}$$

The subset of conditions is then used to calculate the causal effect of g_i on g_t. Conditional on the selected conditions, we search the gene network to find the pathways as described in 2.2.

As our second method, we designed a sampling scheme. Suppose l ($l < M$) conditions are sampled without replacement and denoted as $\mathbf{s}_u$. We re-calculate the correlation coefficient using conditions covered by $\mathbf{s}_u$. $\mathbf{s}_u$ is considered a valid choice of subset if $\xi_{\mathbf{s}_u}(g_t,t_k) > \tau'$, where τ' is a pre-determined threshold for correlation. To make the selection robust and not sensitive to one sample, we repeat the sampling multiple times until we obtain r valid subsets of conditions. The r subsets of conditions ($r \times l$ matrix) is then used to calculate the expected causal effect of g_i on g_t using equation (10) and all the previous equations concerning $\xi(g_i,g_t)$ need to be updated accordingly.

$$\overline{\xi_{\mathbf{s}}}(g_i,t_k) = \frac{1}{r}\sum_{u=1}^{r}\xi_{\mathbf{s}_u}(g_i,t_k). \tag{10}$$

It's obvious that the first method is computationally efficient compared to the second one. However, this method can be heavily affected by outliers and conditions cannot be ''tuned'' for specific TFs. Although the second method is much more time consuming and could fail either because such conditions do not exist or due to extremely large sample space, it's generally more robust and will be much less affected by outliers.

2.4 Significance measurement

It's essential to test the reliability of the inferences by the above approaches. Erroneous inference could be caused purely by false positive interactions in the gene network and the noisy expression data. Therefore, for g^*_c which satisfies (8), the significance of the findings need to be evaluated. As the main source of error comes from the network topology and gene expression, we permute the gene network while preserve the degree of each node similarly as (Milo, *et al*., 2002). Since the network topology is randomized, the pathway-wise expressions are permutated accordingly too. For each permutation, we perform the same procedure and obtain one $V'_T(g^*_c)$. By repeating the permutation many times and ordering all the $V'_T(g^*_c)$, we can calculate an empirical P-value for the $V_T(g^*_c)$. P-values less than 0.05 are considered as significant and those $g^*_c s$ will be reported as valid findings.

3 RESULTS

3.1 Data collection

Rosetta compendium data (Hughes, *et al*., 2000) is used to verify our method. 276 genes were deleted and each deletion mutant's expression profile was measured using microarrays. To build the gene network, the protein-protein interaction data was obtained from a previous compiled set by (Steffen, *et al*., 2002) combined with protein physical interactions deposited in MIPS (Munich Information center for Protein Sequences). TF-DNA binding data was obtained from (Harbison, *et al*., 2004) where 203 TFs were tested for their binding profiles in yeast. We chose P<0.001 as the threshold for positive binding as used by the original authors. The genome wide phosphorylation information was obtained from (Ptacek, *et al*., 2005) which identified over 4,000 phosphorylation events. After compiling all three types of interactions together, the gene network covered 4,744 genes and contained more than 10,500 edges. Our main experiment is based on a recent genome-wide study on expression variation by crossing two yeast strains (Brem, *et al*., 2002; Yvert, *et al*., 2003; Brem, *et al*., 2005). 112 segregants were individually genotyped at 2,956 marker positions and 6,228 gene expressions were measured for each segregant. Since both the genotype and expression of each gene are known, these data are excellent for this study.

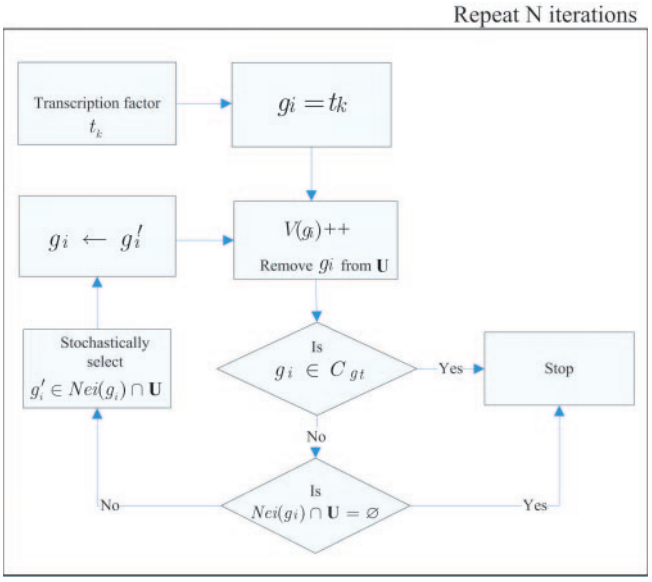


Fig. 3. The flow diagram of the stochastic searching algorithm.

3.2 Testing with knock-out data

In order to quantitatively measure the performance of our method, we designed a test using Rosetta compendium knock-out data. The main reason to use the knock-out data set is because we know what gene was deleted. As the true cause for the gene expression variation is known, we are able to test the accuracy of the inferences. Although the original experiments are not related to eQTL mapping, we can easily define regions around the deleted genes so the problem will be the same as what we are trying to solve. The major steps of the test are described as follows.

(1) For a deletion mutation experiment, we identify the genes whose expression are significantly perturbed. We further identify the common TFs for these significantly perturbed genes, only genes regulated by the common TFs are considered as valid target genes and are used for the later inferences.

(2) We simulate an eQTL region around the deleted gene to let the pretended eQTL contain 10 genes. These 10 genes (the deleted gene and the surrounding 9 genes) position consecutively on

the chromosome. The position of the deleted gene is randomly set to be from 1 to 10.

(3) We pretend that the real causal gene (in this case, the deleted gene) is unknown and try to identify it from the ten genes.

(4) The overall prediction accuracy is calculated. As random guess will give a 10% correct identifications in expectation, higher accuracy is expected given that the method actually works.

For our method to work, it's essential to ensure that the ''differentially'' expressed genes are really caused by deletion mutation instead of by noise. Obviously, a target gene whose expression is perturbed by random events won't lead us to any meaningful findings regardless of the method. Hughes *et al*. designed a gene-specific error model to compensate for the differences in the variation of transcript (Hughes, *et al*., 2000). Based on this error model, more than half of the deletion mutation experiments didn't show significant changes in expression profiles and are excluded from our test. 118 knock out experiments contain at least 2 genes with 3 fold changes with P-value less than 0.01. The number of perturbed genes varieed significantly among these experiments (from 2 to several hundred). We further required that the target genes should share common TFs. By only considering genes that could be clustered by common TFs, we are more confident in believing that their expression variation is caused by the knockout instead of by chance.

We developed a simple voting scheme to consider multiple perturbed genes for each knockout experiment. The genes obtaining the most votes are reported as causal genes. Finally, 17/36 valid predictions are correct (exactly match the deleted gene) using the first condition selection method and 16/35 are correct using the second condition sampling method. The accuracy rates (47% and 46%) are more than 4 times better than what would be expected by random guess. When the eQTL region is set to contain 20 genes, 15 out of 48 predictions are correct by the first method and 15 out of 44 are correct using the second method. The accuracy rates (31% and 34%) are more than six times better compared with random guess (5%). The correct prediction only decreases by two/one when the number of genes in eQTL doubles. This indicates that our method is quite robust and relatively insensitive to the number of genes in eQTL (details and list of genes are provided in the supporting materials). The good performance suggests that our approach can indeed extract useful information from multiple data sources and generate valid hypothesis. We then applied our method to a recent genome-wide study on expression variation in yeast where the causal genes are generally not known (Brem, *et al*., 2005).

3.3 eQTL mapping

We performed 6,228 × 2,956 Wilcoxon ranksum tests to examine the association between each gene's expression level and each marker as in (Brem, *et al*., 2002; Bing and Hoeschele, 2005). We only considered genes whose expression variation could be significantly linked to exactly one locus on yeast genome (P-value<10^{-5}) and the false discovery rate (FDR) was estimated to be 0.005 using methods from (Storey and Tibshirani, 2003). This gave us a list of 1,226 genes. Based on these genes, we performed bootstrap to infer the 95% confidence interval similarly as (Bing and Hoeschele, 2005). A small fraction of genes failed to generate valid confidence intervals and were excluded for further consideration. Finally, we obtained a list of 1,085 genes. The length of the confidence intervals ranges from 781bps to 141Kbps, the mean and the standard deviation are 35Kbps and 28Kbps, respectively. The number of genes within each interval ranges from 1 to 62, and the mean and standard deviation are 16.8 and 15.3, respectively.

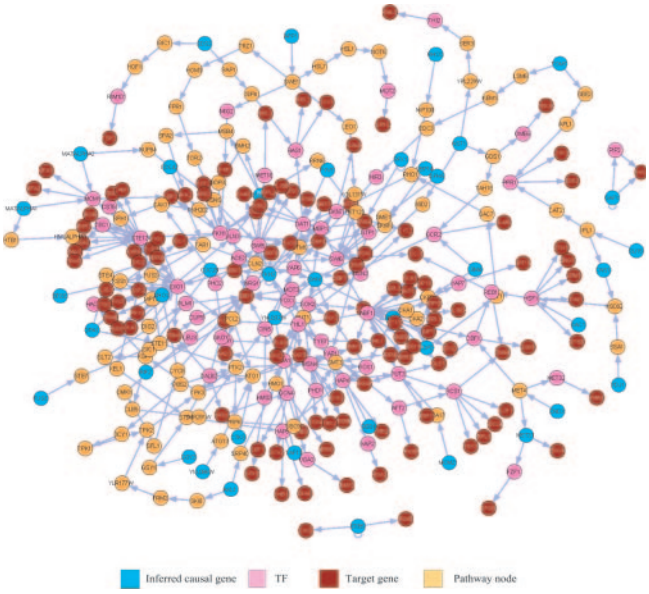


Fig. 4. The global view of the inferred regulatory network. List of all the pathways is provided in the supporting materials.

3.4 Causal gene inference

For the genes in the above list, we applied our algorithm to infer the causal gene in each eQTL. To identify the TF that really involves in the pathway, we require that TF displays a strong correlation with the target gene based on our sub-condition sampling scheme. Clearly, this criterion may be too strong for some TFs and still not sufficient for others. However, the assumption that stronger correlation implies higher probability of regulation could be valid in general. For the 1,085 genes with valid eQTL regions, 585 genes have in total 1,403 highly correlated TFs ($|\rho(g_i, t_k)|$>0.5). For these 585 genes, we inferred the causal genes and measured the significance for them. As described in 2.3, two methods were used to select appropriate subset of conditions. These two methods generated quite similar outputs and we only present the results generated by the second method. 239 inferences have P-value <0.05 and they are reported in supplementary files. The underlying pathways were inferred and are shown in Figure 4, drawn by Cytoscape (Shannon, *et al*., 2003).

Here, we describe two examples which are well supported by experimental data and previous studies. As the first example, the target gene is PRP39, a component of RNA splicing factor U1 small nuclear ribonucleoprotein polypeptide (Lockhart and Rymond, 1994). There's no report on its expression regulatory mechanism by SGD (Saccharomyces Genome Database) (Cherry, *et al*., 1997). Based on the linkage analysis, variation of PRP39's expression can be significantly linked to a locus on chromosome VIII (P-value is 1e-7.3). The 95% confidence interval contains three genes (NEM1, GPA1 and MRS11). From chromatin (Ch) immunoprecipitation (IP) experiments, two TFs (DIG1 and STE12) bind to the promoter

Table 1. Pathway-wise increase of the correlation when appropriate subset of conditions is selected

Genes on the path	Target Gene (PRP39) No subcondition sampling	*Condition on >0.4	*Condition on >0.5	*Condition on >0.6
DIG1	0.27	0.45	0.52	0.62
FUS3	0.50	0.55	0.55	0.60
FAR1	0.49	0.60	0.62	0.64
STE4	0.37	0.52	0.53	0.53
GPA1	0.53	0.61	0.61	0.62

*When a specific condition is set, we require the TF, in this case DIG1 will have a correlation coefficient with the target gene (PRP39) at least or above the threshold under the selected conditions.

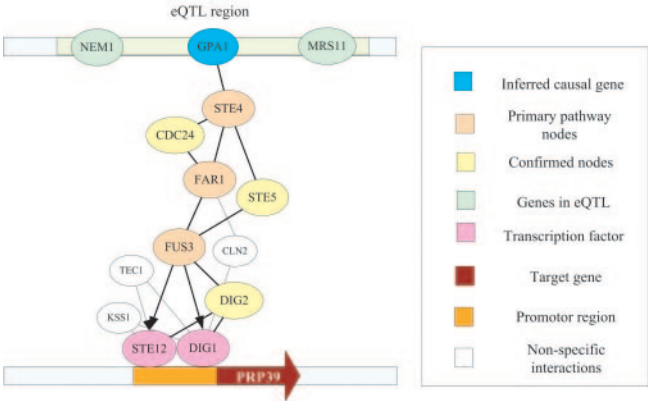


Fig. 5. An example of the inferred causal gene and the pathway. Edges without arrows are protein-protein interactions. Edges with arrows represent phosphorylation or TF-DNA binding. The causal gene is correctly inferred in this example and proteins involved in the pathway are highlighted in colors. Only nodes been visited more frequently than GPA1 and have at least two interactions with primary pathway nodes are shown.

region of PRP39 gene. For each TF, we first sample a subset of conditions so the TF correlates with PRP39 in their expressions. Conditional on these conditions (Table 1), we stochastically search paths which connect TF with the candidate causal genes so that the nodes on the path show significant correlation with PRP39. Two TFs report the same gene (GPA1) as the causal gene as it has the highest probability (0.975) among the three genes with P-value <0.05 by permutation test. The pathways identified by each TF are also consistent. There are quite a few genes (e.g., FAR1, FUS1, etc.) having the same inferred causal gene and pathway. Many of them are known to be involved in pheromone signaling pathway (Wang and Dohlman, 2004). By comparing the pathway we found (Figure 5) with the known pheromone pathway, large fraction of proteins are matched and arranged in a correct order. To further verify that GPA1 was indeed the cause for the downstream gene expression variation, Yvert *et al*. performed experiment by making a point mutation on GPA1 in one of the yeast strain and observed that those downstream genes displayed altered expression levels as expected (Yvert, *et al*., 2003). Here, we show our method can correctly infer the right causal gene and

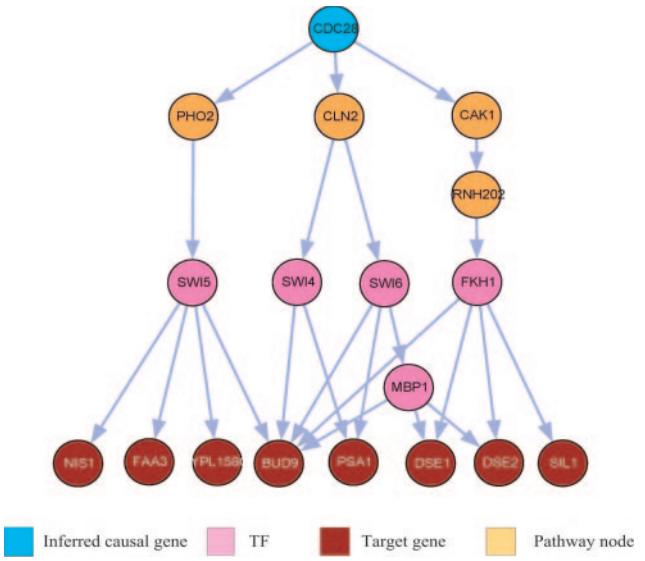


Fig. 6. Inferred pathways related to G1/S phase transition. The arrow only represents the direction of the causal relationship and doesn't stand for phosphorylation or TF-DNA interaction.

derive the underlying pathway without any prior knowledge of the corresponding pathway.

In Table 1, we list the primary nodes on the above pathway and their expression correlations with the target gene. We show both the correlations calculated without selecting a subset of conditions and correlations calculated with subset conditions sampled. The conditions are sampled based on different thresholds and it's clear that as the thresholds increase, the correlations increase accordingly in a pathway-wise manner. This supports the validity of the pathway from a different aspect.

We take G1/S phase transition pathway as our second example. In this example, we identified a group of genes which were reported to be regulated by CDC28. CDC28 is a catalytic subunit of the main cell cycle cyclin-dependent kinase (CDK). The pathways form a complex network and are shown in Figure 6. We list the Gene Ontology annotations of the proteins involved in the pathways in Table 2. It's clear that most genes we inferred are indeed related to the mitotic cell cycle and most interactions and regulations are supported by previous studies.

Deriving complicated networks such as the one shown in Figure 6 could take years by biologists using traditional biology experimental methods. Here, we show that it can be easily obtained by computational methods by integrating multiple sources of high-throughput data. Although computational approach cannot replace the traditional experiments, they do generate valid and testable hypothesis which can help biologists to be more productive.

4 DISCUSSION

We developed a novel approach to estimate the probability for genes in the eQTL of being the causal gene for the target gene expression variation. We show that the causal gene inference problem can be combined with pathway finding problem to achieve a unified solution. Traditionally, genetic studies can only locate a region on

Table 2. Function annotation of the proteins involved in the G1/S phase transition. Descriptions are obtained from GO and SGD database or from referred studies

Gene	Function
Causal gene	
CDC28	Cyclin-dependent protein kinase, involved in G1/S and G2/M transition
Pathway genes	
PHO2	Involved in phosphate metabolism, could interact with CDC28 (Liu, *et al*., 2000)
SWI5	G1 specific TF, bind cooperatively to HO promoter with PHO2 (Bhoite and Stillman, 1998)
CLN2	G1 cyclin, activates CDC28 to promote the G1/S phase transition
SWI4/SWI6/MBP1	Form complexes, regulate transcription at the G1/S transition
CAK1	Kinase, activates CDC28
RNH202	Ribonuclease H activity
FKH1	Regulates the cell cycle and pseudohyphal growth
Target genes	
NIS1	Possibly involved in a mitotic signaling network
FAA3	Long chain fatty acyl-CoA synthetase
YPL158C	Regulated by SWI5 (Doolin, *et al*., 2001), unknown function
BUD9	Involved in bud-site selection
PSA1	Cell wall biosynthesis, required for cell wall structure
DSE1/DSE2	Cell wall organization and biogenesis
SIL1	Molecular function unknown

chromosome which likely contains the causal gene. Our approach digs deep into the biological system to explore the underlying mechanism. We model biological system as a stochastic network of interactions and regulations, the causal effect is explained by the pathways which link the genes in the eQTL and the TFs which potentially regulate the target genes. The eQTL information plays an important role in significantly reducing the number of possible pathways need be considered while pathway identification ultimately helps to answer which gene is the causal gene.

Our methods rely on the network built on protein-protein interaction, TF-DNA binding, and protein phosphorylation. The advantage of this is that the generated pathways can have direct experimental supports. However, none of the above data is either complete or completely accurate (von Mering, *et al*., 2002; Deng, *et al*., 2003). Therefore, important pathways may be missed due to incompleteness of the data and causal genes may be erroneously inferred if it's derived from the inaccurate part of the data. Although we expect to see more abundant and accurate data available in the future, a robust method minimally affected by the data imperfectness is always desired. Compared with deterministic approaches, our method has an inherent stochastic component which makes it resistant to some errors. We intend to further test the robustness of our methods in future work.

As described in the method section, we assume that genes on the pathway will have higher expression correlation with the target genes. This is clearly true for the pheromone pathway which we presented as an example. Moreover, our quantitative test on yeast knock-out experiments indicates this assumption holds for many cases. However, as the biological system is very sophisticated, we don't expect such simple assumption holds for all the cases. Much deeper understanding of the biological regulatory mechanism is needed for a more realistic modeling and we'll improve that in the future.

Gene expression level changes are found common to many diseases such as cancers (Bals and Jany, 2001; van 't Veer, *et al*., 2002; Hauser, *et al*., 2003). Therefore, it will be very interesting to explore on extending our methods to disease causal gene identification (Schadt, *et al*., 2005). Once we identify genes whose expression change significantly between healthy individuals and patients, our approach can be applied to find the genes responsible for these changes. Although the findings may at large be hypothesis by itself, it will significantly improve our understanding of the complex disease scenario by providing a global view of the whole system.

ACKNOWLEDGEMENTS

We thank Drs. Rachel B. Brem and Leonid Kruglyak for kindly providing us the yeast genotype data. This work was inspired by collaboration with Huiying Yang from Medical Genetics Institute at Cedars-Sinai. We also thank Xianghong Jasmine Zhou for her constructive suggestions. We are very grateful to the researchers who performed all the experiments to generate the data that were used by this study. We apologize to those whose works are not cited due to page limit. This research was supported by NIH/NSF joint mathematical biology initiative DMS-0241102 and by NIH P50 HG 002790.

REFERENCES

Bals,R. and Jany,B. (2001) Identification of disease genes by expression profiling. *Eur Respir J*, **18**, 882–889.

Basso,K., Margolin,A.A., Stolovitzky,G., Klein,U., Dalla-Favera,R. and Califano,A. (2005) Reverse engineering of regulatory networks in human B cells. *Nat Genet*, **37**, 382–390.

Bhoite,L.T. and Stillman,D.J. (1998) Residues in the Swi5 Zinc Finger Protein That Mediate Cooperative DNA Binding with the Pho2 Homeodomain Protein. *Mol. Cell. Biol.*, **18**, 6436–6446.

Bing,N. and Hoeschele,I. (2005) Genetical genomics analysis of a yeast segregant population for transcription network inference. *Genetics*, **170**, 533–542.

Brem,R.B., Storey,J.D., Whittle,J. and Kruglyak,L. (2005) Genetic interactions between polymorphisms that affect gene expression in yeast. *Nature*, **436**, 701–703.

Brem,R.B., Yvert,G., Clinton,R. and Kruglyak,L. (2002) Genetic dissection of transcriptional regulation in budding yeast. *Science*, **296**, 752–755.

Cherry,J.M., Ball,C., Weng,S., Juvik,G., Schmidt,R., Adler,C., Dunn,B., Dwight,S., Riles,L., Mortimer,R.K. and Botstein,D. (1997) Genetic and physical maps of Saccharomyces cerevisiae. *Nature*, **387**, 67–73.

Deng,M., Sun,F. and Ting,C. (2003) Assessment of the reliability of protein-protein interaction and protein function prediction. *Pac Symp Biocomput*, **2003**, 140–151.

Dodge-Kafka,K.L., Soughayer,J., Pare,G.C., Carlisle Michel,J.J., Langeberg,L.K., Kapiloff,M.S. and Scott,J.D. (2005) The protein kinase A anchoring protein mAKAP coordinates two integrated cAMP effector pathways. *Nature*, **437**, 574–578.

Doolin,M.-T., Johnson,A.L., Johnston,L.H. and Butler,G. (2001) Overlapping and distinct roles of the duplicated yeast transcription factors Ace2p and Swi5p. *Molecular Microbiology*, **40**, 422–432.

Friedman,N., Linial,M., Nachman,I. and Pe'er,D. (2000) Using Bayesian Networks to Analyze Expression Data. *Journal of Computational Biology*, **7**, 601–620.

Harbison,C.T., Gordon,D.B., Lee,T.I., Rinaldi,N.J., Macisaac,K.D., Danford,T.W., Hannett,N.M., Tagne,J.B., Reynolds,D.B., Yoo,J., Jennings,E.G., Zeitlinger,J.,

Pokholok,D.K., Kellis,M., Rolfe,P.A., Takusagawa,K.T., Lander,E.S., Gifford,D.K., Fraenkel,E. and Young,R.A. (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**, 99–104.

Hauser,M.A., Li,Y.-J., Takeuchi,S., Walters,R., Noureddine,M., Maready,M., Darden,T., Hulette,C., Martin,E., Hauser,E., Xu,H., Schmechel,D., Stenger,J.E., Dietrich,F. and Vance,J. (2003) Genomic convergence: identifying candidate genes for Parkinson's disease by combining serial analysis of gene expression and genetic linkage. *Hum. Mol. Genet*., **12**, 671–677.

Hughes,T.R., Marton,M.J., Jones,A.R., Roberts,C.J., Stoughton,R., Armour,C.D., Bennett,H.A., Coffey,E., Dai,H., He,Y.D., Kidd,M.J., King,A.M., Meyer,M.R., Slade,D., Lum,P.Y., Stepaniants,S.B., Shoemaker,D.D., Gachotte,D., Chakraburtty,K., Simon,J., Bard,M. and Friend,S.H. (2000) Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.

Ideker,T., Ozier,O., Schwikowski,B. and Siegel,A.F. (2002) Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, **18**, S233–240.

Ihmels,J., Friedlander,G., Bergmann,S., Sarig,O., Ziv,Y. and Barkai,N. (2002) Revealing modular organization in the yeast transcriptional network. *Nat Genet*, **31**, 370–377.

Li,H., Chen,H., Bao,L., Manly,K.F., Chesler,E.J., Lu,L., Wang,J., Zhou,M., Williams,R.W. and Cui,Y. (2006) Integrative genetic analysis of transcription modules: towards filling the gap between genetic loci and inherited traits. *Hum. Mol. Genet*., **15**, 481–492.

Li,H., Lu,L., Manly,K.F., Chesler,E.J., Bao,L., Wang,J., Zhou,M., Williams,R.W. and Cui,Y. (2005) Inferring gene transcriptional modulatory relations: a genetical genomics approach. *Hum. Mol. Genet*., **14**, 1119–1125.

Liu,C., Yang,Z., Yang,J., Xia,Z. and Ao,S. (2000) Regulation of the Yeast Transcriptional Factor PHO2 Activity by Phosphorylation. *J. Biol. Chem*., **275**, 31972–31978.

Lockhart,S.R. and Rymond,B.C. (1994) Commitment of yeast pre-mRNA to the splicing pathway requires a novel U1 small nuclear ribonucleoprotein polypeptide, Prp39p. *Mol Cell Biol*, **14**, 3623–3633.

Milo,R., Shen-Orr,S., Itzkovitz,S., Kashtan,N., Chklovskii,D. and Alon,U. (2002) Network Motifs: Simple Building Blocks of Complex Networks. *Science*, **298**, 824–827.

Morley,M., Molony,C.M., Weber,T.M., Devlin,J.L., Ewens,K.G., Spielman,R.S. and Cheung,V.G. (2004) Genetic analysis of genome-wide variation in human gene expression. *Nature*, **430**, 743–747.

Ogawa,N., DeRisi,J. and Brown,P.O. (2000) New components of a system for phosphate accumulation and polyphosphate metabolism in Saccharomyces cerevisiae revealed by genomic expression analysis. *Mol Biol Cell*, **11**, 4309–4321.

Pan,X., Ye,P., Yuan,D.S., Wang,X., Bader,J.S. and Boeke,J.D. (2006) A DNA Integrity Network in the Yeast Saccharomyces cerevisiae. *Cell*, **124**, March 10, 2006.

Ptacek,J., Devgan,G., Michaud,G., Zhu,H., Zhu,X., Fasolo,J., Guo,H., Jona,G., Breitkreutz,A., Sopko,R., McCartney,R.R., Schmidt,M.C., Rachidi,N., Lee,S.-J., Mah,A.S., Meng,L., Stark,M.J.R., Stern,D.F., De Virgilio,C., Tyers,M., Andrews,B., Gerstein,M., Schweitzer,B., Predki,P.F. and Snyder,M. (2005) Global analysis of protein phosphorylation in yeast. *Nature*, **438**, 679–684.

Qi,Y., Ye,P. and Bader,J. (2005) Genetic interaction motif finding by expectation maximization—a novel statistical model for inferring gene modules from synthetic lethality. *BMC Bioinformatics*, **6**, 288.

Rogers,S. and Girolami,M. (2005) A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics*, **21**, 3131–3137.

Schadt,E.E., Lamb,J., Yang,X., Zhu,J., Edwards,S., Guhathakurta,D., Sieberts,S.K., Monks,S., Reitman,M., Zhang,C., Lum,P.Y., Leonardson,A., Thieringer,R., Metzger,J.M., Yang,L., Castle,J., Zhu,H., Kash,S.F., Drake,T.A., Sachs,A. and Lusis,A.J. (2005) An integrative genomics approach to infer causal associations between gene expression and disease. *Nat Genet*, **37**, 710–717.

Scott,J., Ideker,T., Karp,M.R. and Sharan,R. (2005) Efficient algorithms for detecting signaling pathways in protein interaction networks. *Lect Notes Comput SC*, **3500**, 1–13.

Segal,E., Friedman,N., Koller,D. and Regev,A. (2004) A module map showing conditional activity of expression modules in cancer. *Nat Genet*, **36**, 1090–1098.

Segal,E., Shapira,M., Regev,A., Pe'er,D., Botstein,D., Koller,D. and Friedman,N. (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet*, **34**, 166–176.

Shannon,P., Markiel,A., Ozier,O., Baliga,N.S., Wang,J.T., Ramage,D., Amin,N., Schwikowski,B. and Ideker,T. (2003) Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res*., **13**, 2498–2504.

Steffen,M., Petti,A., Aach,J., D'Haeseleer,P. and Church,G. (2002) Automated modelling of signal transduction networks. *BMC Bioinformatics*, **3**, 34.

Storey,J.D. and Tibshirani,R. (2003) Statistical significance for genomewide studies. *PNAS*, **100**, 9440–9445.

Tian,L., Greenberg,S.A., Kong,S.W., Altschuler,J., Kohane,I.S. and Park,P.J. (2005) Discovering statistically significant pathways in expression profiling studies. *PNAS*, **102**, 13544–13549.

Turk,R., t Hoen,P., Sterrenburg,E., de Menezes,R., de Meijer,E., Boer,J., van Ommen,G.-J. and den Dunnen,J. (2004) Gene expression variation between mouse inbred strains. *BMC Genomics*, **5**, 57.

van't Veer,L.J., Dai,H., van de Vijver,M.J., He,Y.D., Hart,A.A.M., Mao,M., Peterse,H.L., van der Kooy,K., Marton,M.J., Witteveen,A.T., Schreiber,G.J., Kerkhoven,R.M., Roberts,C., Linsley,P.S., Bernards,R. and Friend,S.H. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

von Mering,C., Krause,R., Snel,B., Cornell,M., Oliver,S.G., Fields,S. and Bork,P. (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**, 399–403.

Wang,Y. and Dohlman,H.G. (2004) Pheromone signaling mechanisms in yeast: a prototypical sex machine. *Science*, **306**, 1508–1509.

Xing,B. and van der Laan,M.J. (2005) A causal inference approach for constructing transcriptional regulatory networks. *Bioinformatics*, **21**, 4007–4013.

Yeang,C.-H., Ideker,T. and Jaakkola,T. (2004) Physical Network Models. *Journal of Computational Biology*, **11**, 243–262.

Yeang,C.-H., Mak,H.C., McCuine,S., Workman,C., Jaakkola,T. and Ideker,T. (2005) Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biology*, **6**, R62.

Yeger-Lotem,E., Sattath,S., Kashtan,N., Itzkovitz,S., Milo,R., Pinter,R.Y., Alon,U. and Margalit,H. (2004) Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *PNAS*, **101**, 5934–5939.

Yoshimoto,H., Saltsman,K., Gasch,A.P., Li,H.X., Ogawa,N., Botstein,D., Brown,P.O. and Cyert,M.S. (2002) Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in Saccharomyces cerevisiae. *J Biol Chem*, **277**, 31079–31088.

Yvert,G., Brem,R.B., Whittle,J., Akey,J.M., Foss,E., Smith,E.N., Mackelprang,R. and Kruglyak,L. (2003) Trans-acting regulatory variation in Saccharomyces cerevisiae and the role of transcription factors. *Nat Genet*, **35**, 57–64.

Zien,A., Kuffner,R., Zimmer,R., Lengauer, and Thomas (2000) Analysis of gene expression data with pathway scores. *Proc Int Conf Intell Syst Mol Biol*, **8**, 407–417.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e497–e506
doi:10.1093/bioinformatics/btl224

Computational inference of the molecular logic for synaptic connectivity in *C. elegans*

Vinay Varadan[1], David M. Miller III[2] and Dimitris Anastassiou[1,*]

[1]Center for Computational Biology and Bioinformatics (C2B2) and Department of Electrical Engineering, Columbia University, New York, NY 10027, USA and [2]Department of Cell and Developmental Biology, Vanderbilt University, Nashville, TN 37232, USA

ABSTRACT

Motivation: The nematode *C. elegans* is an ideal model organism in which to investigate the biomolecular mechanisms underlying the connectivity of neurons, because synaptic connections are described in a comprehensive wiring diagram and methods for defining gene expression profiles of individual neurons are now available.

Results: Here we present computational techniques linking these two types of information. A systems-based approach (EMBP: Entropy Minimization and Boolean Parsimony) identifies sets of synergistically interacting genes whose joint expression predicts neural connectivity. We introduce an information theoretic measure of the multivariate synergy, a fundamental concept in systems biology, connecting the members of these gene sets. We present and validate our preliminary results based on publicly available information, and demonstrate that their synergy is exceptionally high indicating joint involvement in pathways. Our strategy provides a robust methodology that will yield increasingly more accurate results as more neuron-specific gene expression data emerge. Ultimately, we expect our approach to provide important clues for universal mechanisms of neural interconnectivity.

Contact: anastas@ee.columbia.edu

Supplementary Information: Expression and connectivity data will be available and maintained in the future as new results become available, together with software and additional clarifying descriptions of our techniques, on www.ee.columbia.edu/~anastas/ismb2006

1 INTRODUCTION

Nerve cells (neurons) are interconnected by branching pathways forming complex networks. A fundamental connection mechanism between two neurons is the chemical synapse, a junction by which a presynaptic neuron transfers signals, carried by neurotransmitters, to a postsynaptic neuron. Another connection mechanism is the gap junction, or ''electrical synapse,'' by which ions and small molecules pass from one neuron to the other. Chemical synapses have a well-defined directionality, while gap junctions are bidirectional.

The biological mechanisms governing the selection and formation of synaptic pairs of neurons are not yet well understood. Roger Sperry, in his ''chemoaffinity hypothesis,'' (Sperry, 1963) proposed that synaptic partners express particular combinations of molecular determinants acting as chemical ''identification tags'' that define a productive interaction. Although several candidate molecules have been proposed for this task (see Discussion section), it has been difficult to establish their roles with certainty. One way to infer the molecules responsible for synaptic connectivity would be to analyze the single-cell expression patterns of pairs of neurons known to form synapses. The problem with most nervous systems, however, is that maps of wiring connectivity are not available.

The exception to this rule is the nervous system of the nematode *C. elegans*, which has a simple and well-defined nervous system with only 302 neurons, for which nearly all synaptic connections are described in a comprehensive ''wiring diagram.'' In principle, candidate genes serving as ''synaptic connectivity factors'' for these synapses could be deduced by linking the wiring diagram with the gene expression repertoires of all individual neurons in the network. Until now, most efforts have focused on the interconnectivity of particular neurons only (Miller *et al*., 1992; Winnier *et al*., 1999; Shen *et al*., 2004). However, in our approach, we can now correlate expression data with the entire wiring diagram. Correlation between gene expression and neural connectivity has been previously observed (Kaufman and Ruppin, 2005, personal communication).

In this paper, we develop computational techniques for this task and test them on actual data. Our aim is not classification, which can be achieved using, e.g., SVM-based computational methodology. Rather, it is biological discovery: we seek to infer modules of genes synergistically interacting with each other, which, as expression data become increasingly accurate, will provide insight into related pathways. Because we infer systems of genes rather than individual genes, this methodology is in accordance with the principles of systems biology, and it has the additional feature that it links two different levels of abstraction: The intercellular level of the network of interconnected neurons, as well as the intracellular level of the biomolecular pathways within the neurons.

Furthermore, we introduce an information theoretic measure of the multivariate synergy (section 3.3) and prove that it is exceptionally high in all our results, indicating that the phenotype of synapse formation is the outcome of the interaction of the gene products, rather than from the effect of their individual contributions. This definition of synergy leads naturally to a decomposition of the gene sets, providing further insight into the nature of the mutual interactions among its members.

*To whom correspondence should be addressed.

2 NOTATION AND INPUT DATA

The two types of input data that we link contain the information about connectivity and gene expression. We use the following notation:

1. We refer to the cells (neurons) using the symbol c_i, $i = 1, 2, \ldots, K$, where K is the total number of neurons. The topology of the chemical synapses in the wiring diagram is specified by a $K \times K$ **Adjacency Matrix** $\boldsymbol{A}$, defined so that A_{ij} is 1 if presynaptic c_i connects to postsynaptic c_j with at least one chemical synapse, and 0 otherwise. In other words, $\boldsymbol{A}$ is the adjacency matrix of the directed graph that depicts the wiring diagram. Each branch of the graph corresponds to an oriented chemical synapse, which is defined by an ordered pair (c_i,c_j) for which $A_{ij} = 1$. Gap junctions are electrical synapses and their topology is specified by a different $K \times K$ Adjacency Matrix $\boldsymbol{B}$, which is symmetric, because gap junctions are bidirectional.
2. We refer to the genes for which we have expression data using the symbol g_i, $i = 1, 2, \ldots, M$, where M is the total number of such genes. The genes that are expressed in each neuron are specified by a **Gene Expression Matrix** $\boldsymbol{E}$, defined so that E_{ij} is 1 if g_i is expressed in c_j and 0 otherwise. In other words, $\boldsymbol{E}$ is the gene expression matrix in which each condition corresponds to genes known to be expressed in each neuron. For reasons that we explain below, we assume that the expression data are binary, i.e. the corresponding gene product is either fully present, or absent.

We used connectivity data (Chen *et al*., 2006) recently updated from an earlier version (White *et al*., 1986) for $K = 280$ neurons, resulting in two adjacency matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$, corresponding to chemical synapses and gap junctions respectively.

We extracted single-cell expression data for the gene expression matrix E from the publicly accessible ''Wormbase'' project (Wormbase, 2006) For each of the 280 neurons listed in the connectivity adjacency matrices, we compiled a limited list of genes that are known to be expressed or not expressed in particular neurons as detected by GFP-tagging or antibody experiments. Because graded expression values are generally not available from these results, we created a binary gene expression matrix in which genes are scored as either ''on'' or ''off.'' Using binary expression values also has the benefit of providing sufficient statistics to create the probabilistic models that we use in this paper, and to lead to convenient Boolean logic functions connecting the expression values.

To acquire a list of genes expressed in each of the 280 neurons in our connectivity matrix, we first mined Wormbase using the ''Expression Pattern'' field entry for every *C. elegans* gene. Information in the ''Expression field'' about the tissues or cells in which the gene is expressed is further organized into three sub-fields called ''Summary,'' ''Cell'' and ''Cell Group.'' The ''Cell'' field contains the names of individual cells in which the gene is expressed; the ''Cell Group'' field lists the groups of cells or tissues in which the gene is expressed. We made a list of all the ''Cell Group'' entries available in Wormbase and then manually created a ''translation table'' for each ''Cell Group'' entry related to neurons. This translation table contains a list of neurons that correspond to each cell group entry.

Thus, for each gene, we first compiled all the neurons listed in the ''Summary'' and ''Cell'' fields and then augmented this list with the neurons corresponding to each entry in the ''Cell Group'' field using the translation table. Of the total of 3,363 genes for which expression data are available, we estimated a total of 1,567 genes that are expressed within the nervous system. We further pruned this list of genes by ignoring those genes that are expressed in all neurons (as noted in the ''Cell Group''field) since they do not contribute any information for our purposes. A final labor-intensive task consisted of manually correcting the expression patterns of all remaining genes by checking the information in the referenced papers listed in the Wormbase for each gene, and further removing from the list those genes for which expression data were ambiguous. The final list consisted of $M = 292$ genes.

The 280×280 matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, and the 292×280 matrix $\boldsymbol{E}$ are shown at www.ee.columbia.edu/∼anastas/ismb2006 and we will maintain that site updating these matrices as we obtain more data in the future.

3 ENTROPY MINIMIZATION AND BOOLEAN PARSIMONY

Entropy Minimization and Boolean Parsimony (EMBP) is a systems-based computational methodology that we developed, which identifies, directly from gene expression data, modules of genes (as opposed to individual genes) that are jointly and synergistically associated with a particular outcome, in this case synaptic connectivity. Furthermore, the technique provides insight into the underlying biomolecular logic by inferring a logic function connecting the joint expression levels in a gene module with the outcome. We have recently used the same technique to obtain insight into the disease-related biomolecular logic by analyzing sets of microarray data from diseased and healthy tissues (Varadan and Anastassiou, 2006).

We pose two questions, which are answered sequentially:

(a) Given a number n, identify the set of n genes (subset of the set of all M genes corresponding to the rows of matrix $\boldsymbol{E}$), each of which associated to either the presynaptic or the postsynaptic neuron, whose *joint* expression pattern predicts the existence of a synapse with minimum uncertainty.

(b) Given the above genes, find the simplest logical rule that connects their expression levels to predict the existence of synapses.

Furthermore, we present an information theoretic analysis of the ''synergy'' among these genes with respect to their joint contribution towards synapse formation, which leads to a quantitative measure of synergy and a determination of a decomposition of the gene sets into synergistic modules.

Coupled with additional biological knowledge and possible genetic experimentation, this information can be useful for inferring pathways related to synaptic connectivity. The joint involvement of the members of the gene sets into pathways is supported by the fact that the synergy among them is found to be positive and significantly large.

3.1 Entropy minimization

Addressing the first question, consider the set of all available genes, each of which is counted twice to separately account for its expression in a presynaptic or postsynaptic neuron. Out of this set of size

Table 1. Example of a state-count table.

mig-1 (pre)	*unc-8* (post)	*glr-1* (post)	N_0	N_1	Q/Q_{null}
0	0	0	30472	923	1.05
0	0	1	4412	268	2.05
0	1	0	15334	266	0.61
0	1	1	2491	434	5.30
1	0	0	13641	44	0.11
1	0	1	2031	9	0.16
1	1	0	6571	229	1.20
1	1	1	1254	21	0.59

$2M$, we wish to identify the subset of size n that minimizes the ''uncertainty'' of the existence of a synapse given the gene expression pattern of that subset. It is possible for the optimum subset to contain the same gene twice, which would imply that the formation of a synapse is influenced by its expression in both the pre-synaptic and postsynaptic cell.

We quantify this uncertainty with the information theoretic measure known as conditional entropy, defined as follows (Shannon, 1948). Each of the subsets of size n has 2^n possible gene expression states. For each expression state S, we count the number $N_1(S)$ of times that it is associated with a synapse, and the number $N_0(S)$ of times that it is not associated with a synapse, creating a table with 2^n rows corresponding to the gene expression states, to which we refer as the ''state-count table.'' Each row of the state-count table contains the two counts N_0 and N_1 for the corresponding state. Table 1 shows an example of a state-count table for $n = 3$.

We then create a probabilistic model in which probabilities are equal to relative frequencies derived from the counts N_0 (S) and N_1 (S), so that the presence of a synapse and the gene expression states are random variables. Specifically, we define:

$$P(S) = \frac{N_0(S) + N_1(S)}{K^2}$$

$$Q(S) = \frac{N_1(S)}{N_0(S) + N_1(S)} \text{ if } P(S) > 0$$

The former is the probability of state S in a random ordered pair of neurons and the latter is the probability of synapse given state S. If we know the expression state S of a particular ordered pair of neurons, then the uncertainty of determining whether or not a synapse exists from the first neuron to the second neuron is measured by the entropy $H(Q(S))$, where the function H is defined by $H(q) = -q\log_2(q) - (1-q)\log_2(1-q)$.

The average overall uncertainty of determining whether or not a synapse is formed is then measured by the ''conditional entropy'' of the presence of a synapse given the expression state for the gene set:

$$\sum P(S)H(Q(S))$$

where the summation is over all states S with $P(S) > 0$. The conditional entropy is always a nonnegative number. If it is zero, this implies that the expression state of that subset determines the existence of a synapse with absolute certainty.

More formally, if we use the symbols $G_1, G_2, \ldots, G_n$ for the binary random variables specifying the individual expression states of the n genes defining the joint state S, and the symbol C for the binary random variable specifying the formation of a synapse, then the above conditional entropy is equal to:

$$H(C \mid G_1, G_2, \ldots, G_n) = H(C) - I(G_1, G_2, \ldots, G_n; C)$$

where H is the symbol for the entropy of a random variable, same by convention as the one we used before, and I is the symbol for the mutual information (Cover and Thomas, 1991).

Finally, to ensure that the range of possible values extends from 0 to 1, we normalize the conditional entropy by dividing by $H(C)$, the entropy corresponding to the ''null probability'' Q_{null} of a synapse in a randomly chosen pair of neurons:

$$\frac{H(C \mid G_1, G_2, \ldots, G_n)}{H(C)} = 1 - \frac{I(G_1, G_2, \ldots, G_n; C)}{H(C)}$$

For simplicity, in the sequel we will often refer to the above normalized conditional entropy as just the ''entropy.''

The last column in Table 1 contains, for each state, the relative frequency of a synapse normalized by dividing by Q_{null}. For chemical synapses, Q_{null} is equal to 0.028 (number of synapses divided by K^2). If Q is larger than 0.028, this implies that the expression state contributes favorably towards the creation of a synapse. These evaluations can identify states in which synapses are either overrepresented or underrepresented. For example, in Table 1, synapses are overrepresented in state 011, because the relative frequency of synapses is more than five times larger than the null relative frequency. In other words, neurons that do not express *mig-1* tend to send synapses with increased frequency to neurons that express both *unc-8* and *glr-1*. Another conclusion that we reach from Table 1 is that states 100 and 101 are underrepresented, because, in those states the relative frequencies of synapse are more than six times smaller than the null relative frequency. This means that neurons that express *mig-1* tend to *not* send synapses to neurons that do not express *unc-8*.

If the entropy evaluation is repeated for every possible subset containing n of the $2M$ genes, we can then select the one for which the entropy is minimized. The number of these subsets is equal to $\binom{2M}{n}$ and becomes large for $n \geq 3$, making the exhaustive search method impractical. Therefore, we address this problem using heuristic optimization methods.

We used two different search techniques to determine the minimum entropy gene sets. The first technique starts with a randomly chosen gene set of size n, and iteratively modifies it by replacing one of its genes, chosen at random, with a new gene, also chosen at random from the entire set of $2M$ genes, such that the entropy is minimized. The process is terminated when the entropy has converged. Local minima are avoided by repeating the iterative algorithm with random initial conditions of the same size and select the gene set that yields the overall lowest entropy. This process is repeated for gene sets of size $n + 1$, after ensuring that one of the chosen initial conditions contains the best gene set of size n.

To confirm that the solution is a global minimum, we also used simulated annealing (SA) (Kirkpatrick *et al*., 1983) to search the space of all gene sets of size n. The ''annealing'' process starts at a high ''temperature'' T, corresponding to a disordered system, and slowly cools. The system becomes more ordered at lower temperatures and ''freezes'' at $T = 0$. The search starts with a randomly chosen gene set of size n. A randomly chosen gene in the gene set is replaced by another randomly chosen gene from the

entire set of *2M* genes. If the conditional entropy of the modified gene set is lower than the current gene set, it replaces the current gene set, otherwise, the current gene set is replaced with a probability that is proportional to the temperature at the time (following an exponential cooling scheme) and inversely proportional to the difference between the conditional entropies of the current gene set and the new gene set. Thus, as the temperature falls, ever smaller increases in conditional entropy are accepted, constraining the search only in the local neighborhood of the conditional entropy value.

Rather than ranking genes based on a score measuring the genes' individual contributions to synaptic specificity, this entropy minimization is a systems-based approach attempting to identify modules (sets) of genes in terms of their contribution to jointly and synergistically determine synaptic connectivity. Consequently, the optimum found set of n_1 genes will not necessarily be a subset of the optimum found set of n_1 genes if $n_1 < n_2$.

3.2 Boolean parsimony

Once we have identified the set of genes resulting from entropy minimization, we would like to also infer the ways in which the joint expression levels of these genes determine the resulting phenotype (in our case, synapse formation). We provide two complementary ways of doing this, first (in this section) a technique of determining a simple logic function connecting the individual expression levels, and then (next section) a way to decompose the set into synergistically interacting modules.

The state-count table for the gene set gives us a wealth of information. For example, Table 1 presents the state-count table for the genes that minimize the entropy for $n = 3$, and we have already observed that it indicates that the formation of a synapse is favored if the presynaptic neuron does not express *mig-1*, while the postsynaptic neuron expresses both *unc-8* and *glr-1*. For small values of n, as in the examples shown in the paper, we can label the states for which the relative frequencies of synapses is significantly higher than Q_{null} as ''logic 1'' and use Karnaugh map logic design methodology (Mano, 1979) to identify a simple Boolean function describing the logic under which the phenotype is present. For higher values of n we can use sophisticated algorithms to derive the Boolean function (Brayton *et al*., 1985; Yang and Ciesielski, 2002). The computational problem can be formulated as deriving the ''most parsimonious Boolean function,'' defined as the one minimizing the total number of times of appearance of logic variables connected by the operators AND, OR and NOT. This logic minimization is desirable so that we clarify the biological role for the genes.

Following conventions of Boolean algebra (Boole, 1854) we represent the operator AND as multiplication and the operator OR as addition. For the operator NOT we use the symbol of prime ($'$) following the logic variable. For example, the logic expression ab+a$'$b$'$+ab$'$ is equivalent to the more parsimonious a+b$'$.

3.3 Synergy

This paper introduces an information theoretic measure of multivariate synergy. Because systems biology is based on a holistic view of biological systems, the concept of synergy lies at the heart of it.

Consider a set of n genes with expression levels $G_1, G_2, \ldots, G_n$ and a particular outcome C, which in our case is the formation of a synapse, but it could also be any other phenotype, such as the presence of a particular disease or the differentiation of stem cells into a particular cell type when analyzing expression data of human tissues.

We define the synergy $\text{Syn}(G_1, G_2, \ldots, G_n;C)$ of the gene set with respect to the phenotype C, by:

$$I(G_1, G_2, \ldots, G_n; C) - \max_{\substack{\text{all partitions} \\ \{S_i\}\text{such that} \\ \cup_i S_i = \{G_1, \ldots, G_n\}\text{and} \\ \cap_i S_i = \emptyset}} \sum_i I(\mathcal{S}_i; C)$$

The partition of the gene set that is chosen in the formula above is the one that maximizes the sum of the amounts of mutual information connecting the subsets of that partition with the phenotype, and we will refer to it as the ''synergistic partition'' of the gene set$\{G_1, G_2, \ldots, G_n\}$ with respect to the phenotype C. The definition is naturally consistent with the intuitive concept that synergy is the additional amount of contribution for a particular task provided by an integrated ''whole'' compared with what can best be achieved, after breaking the whole into ''parts,'' by the sum of the contributions of these parts. We may wish to divide the above quantity by the entropy $H(C)$, in which case the maximum possible thus normalized synergy will be +1.

For the special case of $n = 2$, the synergy $\text{Syn}(G_1,G_2;C)$ is equal to $I(G_1, G_2; C) - [I(G_1, C) + I(G_2, C)]$. This measure of bivariate synergy has been previously defined by neuroscience researchers (Gawne and Richmond, 1993; Schneidman *et al*., 2003). In that case, remarkably, it happens to be symmetric with respect to the three random variables and equal to the opposite of the mutual information $I(G_1;G_2;C)$ common to the three variables G_1,G_2,C (McGill, 1955). Contrary to the mutual information common to two variables, the mutual information common to three variables is not necessarily a nonnegative quantity, a fact that is often considered ''unfortunate'' by information theorists (Cover and Thomas, 1991, p. 45). For our purposes, however, this is a fortunate fact, because it allows for strictly positive synergy, as we confirm in our EMBP results, from which we obtain evidence for, and insight into, cooperative participation in biomolecular pathways.

The generalization of the mutual information common to all variables, although elegantly defined in the form of a telescopic sum, has a complicated and not immediately useful physical meaning and cannot be used to properly define the synergy for $n > 2$. A simpler definition of multivariate synergy in the form of $I(G_1, G_2, \ldots, G_n; C) - \sum_{i=1}^{n} I(G_i; C)$ is not appropriate either, as it fails to consider the various ways by which ''parts'' may cooperatively define the ''whole.''

The concept of synergy can be understood by a simple example: Assume that each of the genes G_1 and G_2 is equally (50% of the time) expressed when $C = 1$ and $C = 0$. In that case, it would appear that the two genes are uncorrelated with the phenotype C, because $I(G_1; C) = I(G_2; C) = 0$, and the genes would not be found high up in any typical ''gene ranking'' computational method! However, it is still possible for C to be determined with absolute certainty from the joint state of the two genes, for example when $C = 1$ if $G_1 = G_2$, and $C = 0$ if $G_1 \neq G_2$, in which case $I(G_1, G_2; C) = 1$, and the synergy is positive and equal to +1. On the other hand, if $G_1 = G_2 = C$ then the synergy is negative and equal to -1. More

generally, if $G_1 = G_2 = \cdots = G_n = C$ (ultimate redundancy) then the multivariate synergy can become even more negative and equal to $-(n-1)$.

Since $H(C|G_1, G_2, \ldots, G_n) = H(C) - I(G_1, G_2, \ldots, G_n; C)$, EMBP analysis naturally tends to find high-synergy results, although not necessarily the most synergistic ones (we have the option, if we wish, of modifying the objective function so that we maximize the synergy of the gene set).

Positive synergy implies some form of direct or indirect interaction of all the genes, as a system. An additional advantage of our definition of synergy is that we can obtain insight into the structure of potential pathways (complementary to the insight obtained by performing Boolean parsimony) by making iterative use of the ''synergistic partition,'' defined earlier, to generate a hierarchical decomposition of the gene set into smaller modules. In particular, consider a rooted and not necessarily binary tree with n leaves, each of which represents one of the genes. Each node of the tree represents a subset of genes, which contains the genes represented by the leaves of the clade formed by the node. Therefore the root represents the whole gene set. The synergistic partition of the whole gene set, as defined above, can then be represented by the branching of the root, so that the nodes that are neighboring to the root represent the gene subsets defined by the synergistic partition. Some of these nodes may be leaves, representing a single gene. If they are not leaves, then they represent a subset of genes, which has *its own* synergistic partition, defined and evaluated as above, with respect to the phenotype. This methodology can be repeated for all gene subsets, until the full tree is formed. We refer to this as **the tree of synergy** of the gene set $\{G_1, G_2, \ldots, G_n\}$ with respect to the phenotype C. Each intermediate node of the tree of synergy identifies a gene subset with nonnegative synergy (otherwise our definition of synergy would be contradicted).

The synergy, as defined above, refers to the combined cooperative participation of *all* n genes. If, for example, the expression of one of these genes is independent of all the other genes including the phenotype, then the synergy of the n-gene set will be zero, even if the set contains synergistic subsets. Therefore, for a thorough synergistic analysis of a gene set, we may wish to also identify the most synergistic subsets of size $n-1$, $n-2$, $\ldots$, 2, which may not necessarily appear in the tree of synergy. For $n = 3$, however, it can be easily proved that the most synergistic subset of size 2, if it has positive synergy, is always defined by an existing clade of the full tree of synergy.

For small sizes of gene sets, synergistic analysis can be done with algorithms that list all the partitions of a particular set of genes. The total number of partitions of a set with n elements is given by the Bell number (Kreher and Stinson, 1999). As n increases, however, the increased computational complexity makes the problem intractable and in need of heuristic solutions.

4 RESULTS OF EMBP ANALYSIS

In this section, we apply EMBP analysis using matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{E}$ for *C. elegans*. We ascertain the statistical significance of our results by confirming that the estimated probability that these results would be derived on the basis of pure chance is extremely low. We present the optimum found gene set and the corresponding Boolean logic function for both chemical synapses and gap junctions, using, as an example, a gene set size of $n = 4$.

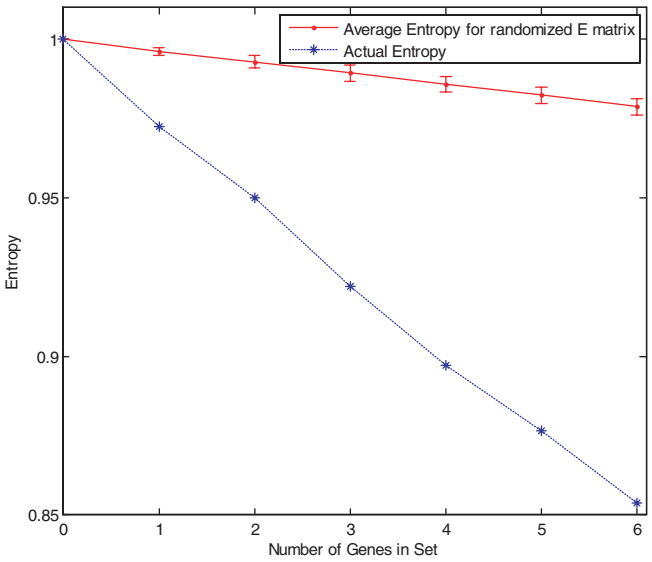


Fig. 1. Comparison of the actual entropies for chemical synapses with those derived from permuted expression matrices averaged over 50 random trials.

4.1 Chemical synapses

To validate that our results are biologically meaningful as opposed to being due to pure chance, we performed entropy minimization using both the actual expression matrix, as well as a number (50) of fictitious expression matrices in which the columns were randomly permuted, so that each neuron is randomly assigned the expression profile of a different neuron. In all cases, we consistently used identical predefined values for all parameters, such as the choice of initial conditions, number of experiments for each gene set size, number of iterations in each step, etc. In this way, all results could be meaningfully compared with each other, because the small probability that a global minimum was missed is identical in all experiments.

Figure 1 shows the minimum normalized conditional entropies for gene set sizes ranging from 0 to 6. The solid red line is derived from the actual expression matrix, while the blue dotted line shows the average of the 50 experiments with permuted expression matrices. The standard deviations of the entropies found in the latter experiments are indicated by the vertical line segments for each value of n.

It is evident from the figure that the entropy minimization algorithm detected real correlation between gene expression in individual neurons and formation of synapses among them. We also observed that the 50 entropy values derived from the permuted expression matrix consistently fit a normal distribution using any of the Chi-squared, Lillie and Geary tests (Walpole *et al*., 2002).

For $n = 4$ we found the following minimum entropy gene set (entropy $= 0.8973$):

a: presynaptic *unc-18*
b: presynaptic *nmr-1*
c: postsynaptic F25B5.2
d: postsynaptic *unc-8*

Given the mean and standard deviation of the entropies and the normality of the distribution, we estimated the P-value, defined as

ab \ cd	00	01	11	10
00	9118 146 0.56	13237 80 0.21	4570 448 3.19	25490 951 1.29
a'cd+bc' / 01	508 68 4.22	703 125 5.39	241 71 8.13	1600 44 0.96
11	0 0 -	0 0 -	0 0 -	0 0 -
10	3577 23 0.23	4966 209 1.44	1933 17 0.31	10263 12 0.04

Fig. 2. The Karnaugh map for the optimum gene set for chemical synapses for $n = 4$. The logic variables a and b defining the rows are presynaptic *unc-18* and *nmr-1*, respectively, and the logic variables c and d defining the columns are postsynaptic F25B5.2 and *unc-8*, respectively.

the probability of obtaining a minimum entropy of 0.8973 or lower on the basis of pure chance, to be 6×10^{-275}.

Figure 2 shows the corresponding Karnaugh map using ''Gray" binary code for easier derivation of the logic function (Mano, 1979), highlighting entries with $Q/Q_{\text{null}} > 2$. Each entry contains the values of N_0, N_1 and Q/Q_{null} (as in a state-count table). The corresponding Boolean function is $a'cd + bc'$ following Karnaugh map methodology and treating the zero-count entries as ''don't care'' states.

In words, these findings are formulated as follows: Neurons that do not express *unc-18* tend to send synapses at higher frequency than normal to neurons that express both F25B5.2 and *unc-8*. Furthermore, neurons that express *nmr-1* tend to send synapses at higher frequency than normal to neurons that do *not* express F25B5.2.

Figure 3 shows the corresponding tree of synergy, where the root and intermediate nodes of the tree are labeled by the normalized synergies of the corresponding gene sets. The quantities within the box are the amounts of normalized mutual information between each gene subset and the formation of synapse, using compact symbols for convenience. For example, $I_{\text{acd}} = (I(a, c, d; C))/(H(C)) = 0.070$. It is instructive to use these numbers to confirm the synergy values at the nodes of the tree. For example, the synergy of the 3-gene set {a, c, d} is evaluated as +0.020, equal to:

$$I_{\text{acd}} - \max \begin{cases} I_{\text{a}} + I_{\text{cd}} \\ I_{\text{c}} + I_{\text{ad}} \\ I_{\text{d}} + I_{\text{ac}} \\ I_{\text{a}} + I_{\text{c}} + I_{\text{d}} \end{cases} = I_{\text{acd}} - (I_{\text{d}} + I_{\text{ac}}) = 0.070 - (0.004 + 0.046)$$

The Boolean functions for the smaller subsets defined by the intermediate nodes provide further insight into the nature of

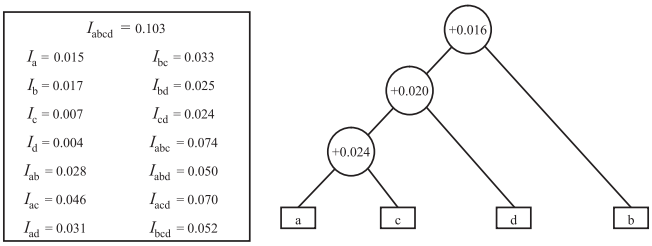


Fig. 3. The tree of synergy for the optimum gene set for chemical synapses for $n = 4$. The leaves correspond to a: presynaptic *unc-18*, b: presynaptic *nmr-1*, c: postsynaptic F25B5.2, d: postsynaptic *unc-8*. See text for additional explanations.

potential gene interactions: It turns out that they are, a′c, a′cd, a′cd+bc′.

To validate our results, we used the same permutations as in Figure 1 and we calculated the synergy for the minimum entropy gene sets for $n = 4$. We also confirmed that the results consistently fit a normal distribution with mean and standard deviation both equal to 0.001. The difference between the actual synergy value of +0.016 and the mean is therefore 15 times the standard deviation, corresponding to an extremely low probability that it is due to pure chance.

Following is a summary of the main properties of the identified genes, three of which (*unc-18*, *nmr-1* and *unc-8*) are already known to encode synaptic components.

UNC-18 and its vertebrate homologs facilitate synaptic vesicle release and are presynaptically localized (Richmond and Broadie, 2002). In *C. elegans*, anti-UNC-18 stains all ventral cord motor neurons, plus additional neurons in the head and tail (Gengyo-Ando *et al.*, 2003).

UNC-8, a DEG/ENaC cation-selective channel subunit is expressed in motor neurons, sensory neurons and interneurons adjacent to the nerve ring (Tavernarakis *et al.*, 1997). In touch neurons, DEG/ENaC channels are believed to function as mechanosensitive transducers (O'Hagan *et al.*, 2005). UNC-8 has been proposed to perform a related function as a stretch receptor in ventral cord motor neurons (Tavernarakis *et al.*, 1997). Recent results strengthen our case that UNC-8 is indeed involved in synaptogenesis (Kawano *et al.*, 2005, personal communication).

nmr-1 encodes an NMDA-type ionotropic glutamate receptor subunit and is expressed in a subset of neurons in the head region including command interneurons that drive motor neuron activity (Francis *et al.*, 2003). In mammals, NMDA-type receptors modulate excitatory postsynaptic responses to glutamate. This activity can result in prolonged changes in synaptic structure and function (Cull-Candy *et al.*, 2001). Synaptic plasticity is also sensitive to an EphrinB signal from the presynaptic membrane that promotes association of the EphB and NMDA receptors (Dalva *et al.*, 2000).

In each of these cases, the implicated proteins are involved in some aspect of synaptic assembly or signaling and thus are plausible candidates having distinct roles in synaptic specificity.

Expression of F25B5.2 appears to be restricted to early embryonic cells of the AB lineage, which later gives rise largely to neurons. F25B5.2 is not expressed in neurons that arise after hatching during larval development (WormBase, 2006). It is intriguing that F25B5.2 appears to also be implicated, in a different way, in the formation of neuron-specific gap junctions (see next section). This

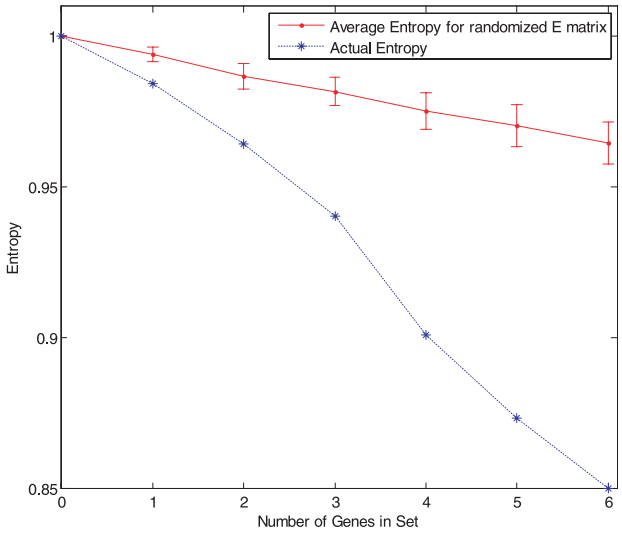


Fig. 4. Comparison of the actual entropies for gap junctions with those derived from permuted expression matrices averaged over 50 random trials.

discovery suggests the possibility that the creation of both electrical and chemical synapses may be coordinated by a common molecular mechanism. Indeed, a unified model is also suggested by the finding that the UNC-4 homeodomain transcription factor orchestrates neuron-specific assembly of gap junctions as well as chemical synapses in the larval ventral cord motor circuit (White *et al*., 1992; Miller and Niemeyer, 1995).

4.2 Gap junctions

Figure 4 shows the normalized conditional entropies for values of n ranging from 0 to 6 for both the actual and the permuted expression matrices, using methodology identical to that described for chemical synapses. Again, it is evident from the gap of several standard deviations between the average entropy values and the actual entropy values that the entropy minimization algorithm extracted biologically relevant information.

For $n = 4$, as was the case for chemical synapses, we affirmed that the conditional entropy values estimated over the 50 random experiments fit a normal distribution, which led to the estimation of the probability of obtaining the minimum found conditional entropy of 0.9010 using the actual $\boldsymbol{E}$ matrix on the basis of pure chance to be 2×10^{-35}. Specifically, we found the following minimum entropy gene set:

a: presynaptic F25B5.2
b: presynaptic *unc-6*
c: postsynaptic F25B5.2
d: postsynaptic *unc-6*

Figure 5 shows the corresponding Karnaugh map, where each contains the values of N_0, N_1 and Q/Q_{null} There are five entries highlighted with bold borders in which the relative frequency of gap junctions is significantly higher than the null frequency. The corresponding Boolean function is:

$$\text{abc}'+\text{a}'\text{cd}+\text{bd}$$

In words, this finding is formulated as follows: Neurons that express both F25B5.2 and *unc-6* tend to form gap junctions at higher frequency than normal with neurons that do not express

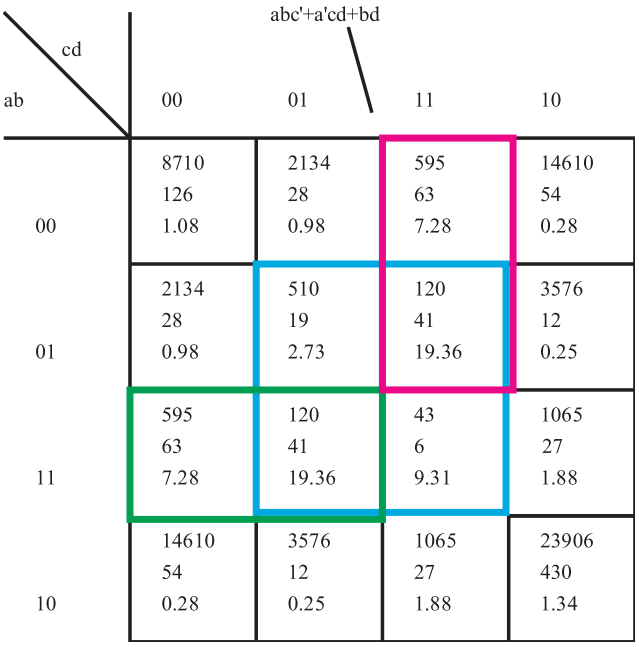


Fig. 5. The Karnaugh map for the optimum gene set for gap junctions for $n = 4$. Each entry contains the values of N_0, N_1, and Q/Q_{null}.

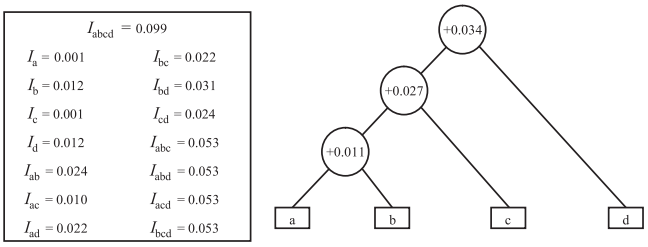


Fig. 6. The tree of synergy for the optimum gene set for gap junctions for $n = 4$. The leaves correspond to a: presynaptic F25B5.2, b: presynaptic *unc-6*, c: postsynaptic F25B5.2, d: postsynaptic *unc-6*. See text for additional explanations.

F25B5.2. Furthermore neurons that express *unc-6* tend to form junctions with each other at higher frequency than normal.

Figure 6 shows the tree of synergy for this gene set, constructed in an identical manner to the tree of Figure 3. A symmetric tree (substituting a for c and b for d) is equivalent. Remarkably, the synergy (+0.034) of the whole gene set is particularly high: more than a third of the mutual information (+0.099) between the gene set and the formation of gap junctions is due to the synergy (+0.034) among these four genes. The Boolean functions for the subsets of the intermediate nodes are ab and abc′.

We validated our results as before confirming that the synergy values for the permuted data fit a normal distribution with mean and standard deviation also both equal to 0.001, making the actual synergy value of +0.034 even more unlikely to be due to pure chance.

Although local gap junction networks are commonly observed in nervous systems, molecular mechanisms that govern the creation

of these electrical connections between specific neurons are unknown (Hestrin and Galarreta, 2005).

unc-6 encodes a guidance cue (Netrin) that regulates axon trajectory and cell migration (Hedgecock *et al*., 1990). Originally discovered in *C. elegans* to steer pioneer axon outgrowth along circumferential tracks, UNC-6/Netrin also performs this function in the axial nerve cords of mammals and insects (Ishii *et al*., 1992; Mitchell *et al*., 1996; Serafini *et al*., 1996). This conserved role is believed to depend on secretion of UNC-6 from selected neurons and ectodermal cells located at the ventral midline (Wadsworth *et al*., 1995; Kennedy *et al*., 1994). Responding neurons express specific UNC-6 membrane receptors, UNC-5 and UNC-40/DCC (Dickson, 2002). Because UNC-6 action in this mode determines the proximity of potential synaptic partners, it clearly imposes at least an indirect effect on the creation of gap junctions between specific neurons. In addition, as a secreted molecule, UNC-6 can exert this role at some distance from the target cell. In the instance considered here, however, UNC-6 expression in adjacent neuronal processes, in concert with F25B5.2, is strongly correlated with gap junction formation. This finding could be indicative of a potentially new role for this potent signaling molecule.

As noted earlier, F25B5.2 is broadly expressed in the embryo and in precursor cells giving rise to a majority of embryonic neurons. Exclusive expression of F25B5.2 in the embryo is correlated with the observed preferential formation of gap junctions between embryonic neurons that do express F25B5.2 and larval neurons that do not. For example, the command interneurons AVAL and AVAR are generated in the embryo but make gap junctions with VA motor neurons that arise during the first larval stage; both of these neuron classes also express UNC-6 (Wadsworth *et al*., 1995). Although the yet undefined function of the F25B5.2 protein does not inform a molecular model of its mode of action, the observation that neurons expressing F25B5.2 in one developmental period (i.e., embryo) are likely to establish gap junctions with neurons that do not express F25B5.2 at a later developmental stage (i.e. larvae) provides a simple paradigm of how temporal expression of other potential determinants may control synaptic specificity.

The exceptionally high value of synergy among F25B5.2 and UNC-6, combined with the facts that F25B5.2 is expressed in neuronal precursors and that UNC-6 creates a hierarchy of netrin cues in the developing nervous system gives rise to the intriguing speculation that these two molecules somehow interact with each other during development with respect to gap junction formation.

5 OTHER COMPUTATIONAL APPROACHES

In its simplest interpretation, Sperry's chemoaffinity hypothesis may be realized in the form of certain ordered pairs of expressed genes in two neurons responsible for the formation of synaptic interconnections. Although in reality this is too simple to be the case, this assumption can still be useful for a computational technique identifying potential synaptic connectivity factors. In this section, we rank all ordered pairs of genes according to a numerical score defining the ''degree of fitness'' to being such factors. In other words, we identify overrepresented gene pairs in pre- and post-synaptic neurons.

For example, we assume that a particular such ordered pair$(g_m, g_n)$of genes expresses heterophilic receptors such that synapses are formed connecting presynaptic neurons expressing gene g_{m} with postsynaptic neurons expressing gene g_n. Resulting synapses ''match'' the ordered pair(g_m, g_n), where we use the term ''A synapse (c_i, c_j) 'matches' an ordered pair of genes(g_m, g_n)'' to indicate that $E_{mi} = E_{nj} = 1$, i.e., that gene g_m is expressed in cell c_{i} and gene g_{n} is expressed in cell c_{j}.

We define the $M \times M$matrix:

$$W = EAE^T$$

Note that element W_{mn} of the matrix $\boldsymbol{W}$:

$$W_{mn} = \sum_{i=1}^{K} \sum_{j=1}^{K} A_{ij} E_{mi} E_{nj}$$

is the total number of synapses that match a particular ordered pair of genes, (g_m, g_n) where m is not necessarily different from n.

The probability P_{actual}, estimated as relative frequency, that a synapse chosen at random will match an ordered pair (g_m, g_n) of genes is:

$$P_{\text{actual}} = \frac{\text{Number of synapses that match } (g_m, g_n)}{\text{Total number of synapses}} = \frac{W_{mn}}{\sum_{i=1}^{K} \sum_{j=1}^{K} A_{ij}}$$

The above probability can be calculated for each ordered pair of genes, but cannot be used as a desired numerical score, because it is biased in favor of the overrepresented ordered pairs of genes, even if such pairs are unrelated to synapses. For it to be used as a relevant numerical score, it must be normalized by dividing by another probability, P_{null} of a ''null'' model, calculated after removing all influence related to gene expression of particular genes. For the null model we assume that we only know the number L_i of neurons expressing each gene g_i:

$$L_i = \sum_{j=1}^{K} E_{ij}$$

The probability that a neuron chosen at random expresses gene g_i is given by:

$$\frac{\text{Number of neurons expressing } g_i}{\text{Total number of neurons}} = \frac{L_i}{K}$$

Thus, according to the null model, the probability that a synapse (or any ordered pair of neurons) chosen at random will match (g_m, g_n) is equal to:

$$P_{null} = \begin{cases} \cong \dfrac{L_m L_n}{K^2} & \text{if } m \neq n \\ = \dfrac{L_n (L_n - 1)}{K(K-1)} & \text{if } m = n \end{cases}$$

In the above formula, the former term results from the assumption that the events of genes g_m and g_n being expressed in the first and second neuron, respectively, are nearly independent of each other and therefore we can use the product of the two probabilities. The latter term is derived by using Bayes' rule, as we know that if gene g_n is expressed in one neuron, then, among the remaining $K - 1$ neurons, the number of them expressing the same gene is $L_n - 1$. It is possible to derive precise formulas, and to improve the null model by making use of the knowledge of the number of genes expressed in each neuron, but these improvements will add complexity without significantly improving relevance.

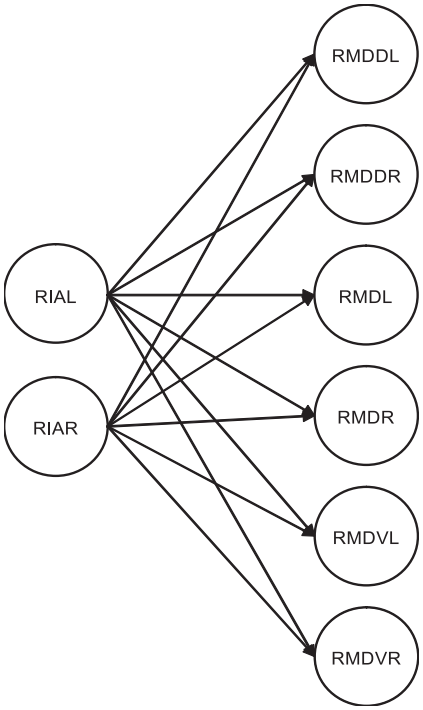


Fig. 7. The set of synapses characterized by the gene-pair (*glr-6*, *rig-5*).

We define the ''log-odds ratio'':

$$LogOdds(g_m, g_n) = \log_2\left(\frac{P_{actual}}{P_{null}}\right)$$

as the numerical score (measured in bits) to rank ordered pairs(g_m, g_n) of potential presynaptic and postsynaptic genes according to the likelihood that they contribute to synapse formation.

Using the augmented expression matrix, we can estimate the above *LogOdds* measure for all the ordered pairs of genes that correspond with the existence of chemical synapses. The gene pairs that achieved the highest score (5.16 bits) with the highest number of connections (12) were (*glr-6*, *rig-5*) and (*glr-3*, *rig-5*) where the first element in each pair is expressed in a presynaptic cell, while the second element is expressed in a postsynaptic neuron. According to the expression matrix, genes *glr-6* and *glr-3* are expressed in the same set of neurons, which are RIAL and RIAR, and gene *rig-5* is expressed in six neurons, which are RMDDL, RMDDR, RMDL, RMDR, RMDVL, and RMDVR. It turns out that each of the former neurons forms a synapse to each of the latter neurons, accounting for a total of $2\times6 = 12$ synapses. In other words, this is an example (Figure 7) of a case in which, without any exception, the pair of genes (e.g., *glr-6* and *rig-5*) determines a chemical synapse, which would suggest that, perhaps in some indirect manner, these genes influence synaptic specificity. Interestingly, *rig-5* encodes a member of the immunoglobulin superfamily of Cell Adhesion Molecules (CAMs) which includes candidate synaptic specificity determinants (Shen, 2004).

Other computational approaches can also be used to infer synaptic connectivity factors. For example, we may wish to address the question: ''Given a particular neuron, what is the gene expression pattern shared by all members of its postsynaptic cluster, defined as the set of its postsynaptic neurons?'' In other words, what is the property that the neuron ''seeks'' in its postsynaptic partner neurons? EMBP analysis can also be used to answer such questions.

6 DISCUSSION

The computational approach of entropy minimization and Boolean parsimony, presented in this paper, is designed to identify modules of synergistically related genes that are correlated with synapse formation. We believe that our strategy, which is designed to identify groups of proteins that together specify synaptic determinants, embodies the fundamental biological complexity of this key event and is therefore more likely to define the molecular underpinnings of synaptic choice than are approaches that seek single genes with this function.

To detect such modules, it is important that a rich set of genes is included in the input data. Our results are severely limited, however, by the small number of genes (1–2% of the predicted genes) with accurate neuron-specific expression patterns currently available in WormBase. In the future, we expect to overcome this limitation by exploiting new microarray-based methods for obtaining gene expression profiles of specific *C. elegans* neurons (Fox *et al*., 2005; Kunitomo *et al*., 2005; Von Stetina *et al*., unpublished data). The cell-type specific expression of genes used in this paper was largely defined by observations of adult animals. As the creation of the nervous system is a dynamic process with active construction underway during both embryonic and larval stages, temporal patterns of gene expression obtained during these critical periods may be especially informative. When utilized with whole genome tiling arrays (Cheng *et al*., 2005), microarray profiling offers the additional benefit of detecting differential expression of alternatively spliced transcripts. These data may be particularly important to our goal of identifying authentic synaptic specificity genes as accumulating evidence indicates that alternative splicing may control neural connectivity. For example, in mammals, various isoforms of cadherins as well as DSCAMs and neurexins have been implicated as synaptic connectivity factors (Cline, 2003; Wojtowicz *et al*., 2004; Wu *et al*., 2001).

Despite these limitations, the correlation that we have found between the wiring diagram and the existing expression data is remarkable, as evidenced both by the validation results and by the high levels of observed multivariate synergy (Figures 1–6). These results establish that the identified expression of combinations of certain gene sets is correlated with synapse formation, although the cause-and-effect relationship between the two events is still unclear. Potential functions for these molecules in synapse formation can be readily tested in *C. elegans* by wedding the power of nematode genetics to an emerging suite of GFP-labeled marker proteins for visualizing synapse formation between specific neurons (Nonet, 1999; Shen *et al*., 2004).

In the future, computational techniques presented here will need to be adjusted to accommodate the substantially increased amount of input data arising from neuron-specific microarray experiments. For example, new more efficient algorithms will be needed to deal with the increased resulting complexity of this analysis. It may also be useful to devise computational approaches in which relative

levels of gene expression are considered rather than the simple binary ''on vs. off'' treatment we have employed in this study.

Once the expression matrix becomes sufficiently enriched and the computational methodology is enhanced to address these challenges, we expect to derive more accurate correlations of specific gene clusters and their alternatively spliced transcripts with synaptic connectivity. The molecular logic of the biological pathways suggested by these data can then be experimentally tested *in vivo*. This approach is expected to identify key genetic mechanisms for regulating synaptic specificity in *C. elegans*. In turn, the results of our work with this simple model organism should reveal valuable clues for the interconnectivity of neurons in more complex nervous systems in which homologous mechanisms are employed to select synaptic partners.

ACKNOWLEDGEMENTS

Part of this work was supported by NIH grant R01 NS26115 to D.M.M.

REFERENCES

Boole,G. (1854) *An Investigation of the Laws of Thought on Which Are Founded the Mathematical Theories of Logic and Probabilities*. London Dover Advanced Mathematics.

Brayton,R.K. *et al*. (1985) *A Logic minimization algorithms for VLSI minimization* Kluwer.

Chen,B.L *et al*. (2006) Wiring optimization can relate neuronal structure and function. *Proc. Natl. Acad. Sci. USA*, **103**, 4723–4728.

Cheng,J. *et al*. (2005) Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science*, **308**, 1149–1154.

Cline,H. (2003) Sperry and Hebb: oil and vinegar? *Trends Neurosci*, **26**, 655–661.

Cover,T.M. and Thomas,J.A. (1991) *Elements of Information Theory*. New York Wiley.

Cull-Candy,S. *et al*. (2001) NMDA receptor subunits: diversity, development and disease. *Curr. Opin. Neurobiol*., **11**, 327–335.

Dalva,M.B. *et al*. (2000) EphB receptors interact with NMDA receptors and regulate excitatory synapse formation. *Cell*, **103**, 945–956.

Dickson,B.J. (2002) Molecular mechanisms of axon guidance. *Science*, **298**, 1959–1964.

Fox,R.M. *et al*. (2005) A gene expression fingerprint of *C. elegans* embryonic motor neurons. *BMC Genomics*, **6**, 42–65.

Francis,M.M. *et al*. (2003) Bridging the gap between genes and behavior: recent advances in the electrophysiological analysis of neural function in *Caenorhabditis elegans*. *Trends Neurosci*., **26**, 90–99.

Gawne,T. and Richmond,B. (1993) How independent are the messages carried by adjacent inferior temporal conrical neurons? *J. Neurosci*., **13**, 2758–2771.

Gengyo-Ando,K. *et al*. (1993) The *C. elegans unc-18* gene encodes a protein expressed in motor neurons. *Neuron*, **11**, 703–711.

Hedgecock,E.M. *et al*. (1990) The *unc-5*, *unc-6*, and *unc-40* genes guide circumferential migrations of pioneer axons and mesodermal cells on the epidermis in *C. elegans*. *Neuron*, **2**, 61–85.

Hestrin,S. and Galarreta,M. (2005) Electrical synapses define networks of neocortical GABAergic neurons. *Trends Neurosci*., **28**, 304–309.

Ishii,N. *et al*. (1992) UNC-6, a laminin-related protein, guides cell and pioneer axon migrations in *C. elegans*. *Neuron*, **9**, 873–881.

Kaufman,A. and Ruppin,E. (2005) Gene Expression, Connectivity and Neural Contributions: A Bridge Too Far? Poster presentation, ISMB.

Kawano,T. *et al*. (2005) A genetic screen for genes affecting synaptogenesis using SYD-2:GFP active zone marker. *International Worm Meeting*, 706C.

Kennedy,T.E. *et al*. (1994) Netrins are diffusible chemotropic factors for commissural axons in the embryonic spinal cord. *Cell*, **78**, 425–435.

Kirkpatrick,S. *et al*. (1983) Optimization by Simulated Annealing. *Science*, **220**, 671–680.

Kreher,D.L. and Stinson,D.R. (1999) Combinatorial Algorithms. CRC Press.

Kunitomo,H. *et al*. (2005) Identification of ciliated sensory neuron-expressed genes in *Caenorhabditis elegans* using targeted pull-down of poly(A) tails. *Genome Biol*., **6**, R17.1–R17.13.

Mano,M.M. (1979) *Digital Logic, Computer Design*. Englewood Cliffs Prentice-Hall, .

McGill,W.J. (1955) Multivariate Information Transmission. *IRE Trans. Info Th*., **4**, 93–111.

Miller,D.M.,III and , Niemeyer,C.J. (1995) Expression of the *unc-4* homeoprotein in *Caenorhabditis elegans* motor neurons specifies presynaptic input. *Development*, **121**, 2877–2886.

Miller,D.M.,III *et al*. (1992) *C. elegans unc-4* gene encodes a homeodomain protein that determines the pattern of synaptic input to specific motor neurons. *Nature*, **355**, 841–845.

Mitchell,K.J. *et al*. (1996) Genetic analysis of Netrin genes in Drosophila: Netrins guide CNS commissural axons and peripheral motor axons. *Neuron*, **17**, 203–215.

Nonet,M.L. (1999) Visualization of synaptic specializations in live *C. elegans* with synaptic vesicle protein-GFP fusions. *J. Neurosci. Methods*, **89**, 33–40.

O'Hagan,R *et al*. (2005) The MEC-4 DEG/ENaC channel of *Caenorhabditis elegans* touch receptor neurons transduces mechanical signals. *Nat. Neurosci*., **8**, 43–50.

Richmond,J.E. and Broadie,K.S. (2002) The synaptic vesicle cycle: exocytosis and endocytosis in Drosophila and *C. elegans*. *Curr. Opin. Neurobiol*., **12**, 499–507.

Schneidman,E. *et al*. (2003) Synergy, Redundancy, and Independence in Population Codes. *J. Neurosci*., **23**, 11539–11553.

Serafini,T. *et al*. (1996) Netrin-1 is required for commissural axon guidance in the developing vertebrate nervous system. *Cell*, **87**, 1001–1014.

Shannon,C.E. (1948) A mathematical theory of communication. *Bell Sys Tech. J*., **27**, 379–423, and 623–656.

Shen,K. (2004) Molecular mechanisms of target specificity during synapse formation. *Curr. Opin. Neurobiol*., **14**, 83–88.

Shen,K. *et al*. (2004) Synaptic specificity is generated by the synaptic guidepost protein SYG-2 and its receptor, SYG-1. *Cell*, **116**, 869–881.

Sperry,R.W. (1963) Chemoaffinity in the orderly growth of nerve fiber patterns and connections. *Proc. Natl. Acad. Sci. USA*, **50**, 703–709.

Tavernarakis,N. *et al*. (1997) *unc-8*, a DEG/ENaC family member, encodes a subunit of a candidate mechanically gated channel that modulates *C. elegans locomotion*. *Neuron*, **18**, 107–119.

Varadan,V. and Anastassiou,D. (2006) Inference of Disease-Related Molecular Logic from Systems-Based Microarray Analysis. *PLoS Comput. Biol*., in press.

Wadsworth,W.G. *et al*. (1995) Neuroglia and pioneer axons express UNC-6 to provide global and local netrin cues for guiding migrations in *Caenorhabditis elegans*. *Neuron*, **16**, 35–46.

Walpole,R.E. *et al*. (2002) *Probability, Statistics for Engineers and Scientists*. New Jersey Prentice Hall, .

White,J.G. *et al*. (1992) Mutations in the *Caenorhabditis elegans unc-4* gene alter the synaptic input to ventral cord motor neurons. *Nature*, **355**, 838–841.

White,J.G. *et al*. (1986) The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philos. Trans. R. Soc. Lond., B, Biol. Sci*., **314**, 1–340.

Winnier,A.R. *et al*. (1999) UNC-4/UNC-37-dependent repression of motor neuron-specific genes controls synaptic choice in *Caenorhabditis elegans*. *Genes Dev*., **13**, 2774–2786.

Wojtowicz,W.M. *et al*. (2004) Alternative splicing of Drosophila Dscam generates axon guidance receptors that exhibit isoform-specific homophilic binding. *Cell*, **118**, 619–633.

Wormbase website. (2006) http://www.wormbase.org, WS155.

Wu,Q. *et al*. (2001) Comparative DNA sequence analysis of mouse and human protocadherin gene clusters. *Genome Res*., **11**, 389–404.

Yang,C. and Ciesielski,M. (2002) BDS: A BDD-based Logic Optimization System. *IEEE Trans. Comp. Design Int. Circ. Sys*., **21**, 866–876.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e507–e513
doi:10.1093/bioinformatics/btl214

Novel Unsupervised Feature Filtering of Biological Data

Roy Varshavsky[1,*], Assaf Gottlieb[2], Michal Linial[3] and David Horn[2]

[1]School of Computer Science and Engineering, The Hebrew University of Jerusalem 91904, Israel, [2]School of Physics and Astronomy, Tel Aviv University 69978, Israel and [3]Department of Biological Chemistry, Institute of Life Sciences, The Hebrew University of Jerusalem 91904, Israel

ABSTRACT

Motivation: Many methods have been developed for selecting small informative feature subsets in large noisy data. However, unsupervised methods are scarce. Examples are using the variance of data collected for each feature, or the projection of the feature on the first principal component. We propose a novel unsupervised criterion, based on SVD-entropy, selecting a feature according to its contribution to the entropy (CE) calculated on a leave-one-out basis. This can be implemented in four ways: simple ranking according to CE values (SR); forward selection by accumulating features according to which set produces highest entropy (FS1); forward selection by accumulating features through the choice of the best CE out of the remaining ones (FS2); backward elimination (BE) of features with the lowest CE.

Results: We apply our methods to different benchmarks. In each case we evaluate the success of clustering the data in the selected feature spaces, by measuring Jaccard scores with respect to known classifications. We demonstrate that feature filtering according to CE outperforms the variance method and gene-shaving. There are cases where the analysis, based on a small set of selected features, outperforms the best score reported when all information was used. Our method calls for an optimal size of the relevant feature set. This turns out to be just a few percents of the number of genes in the two Leukemia datasets that we have analyzed. Moreover, the most favored selected genes turn out to have significant GO enrichment in relevant cellular processes.

Abbreviations: Singular Value Decomposition (SVD), Principal Component Analysis (PCA), Quantum Clustering (QC), Gene Shaving (GS), Variance Selection (VS), Backward Elimination (BE)

Contact: royke@cs.huji.ac.il

Conflicts of Interest: not reported

1 INTRODUCTION

Feature selection is an important tool in many biological studies. Given the large complexity of biological data, e.g. the number of genes in a microarray experiment, one naturally looks for a small subset of features (e.g. small number of genes) that may explain the properties of the data that are being investigated. This type of motivation fits into the general scheme of **feature exploration**, i.e. searching for features because of their direct biological relevance to the problem. An alternative motivation is that of **preprocessing**: searching for a small set of features to simplify computational constraints, to allow for the handling of high throughput biological experiments, and to separate signal from noise. Practically, selection of a small set of genes is of ultimate importance when a small set of informative genes can be the basis for cancer diagnosis and a basis for development of gene associated therapy.

Preprocessing often involves some operation on feature-space in order to reduce the dimensionality of the data. This is referred to as **feature extraction,** e.g. restricting oneself to the first r principal components of a PCA routine. Note that superpositions of features appear in this example. Alternatively, in **feature selection** we limit ourselves to particular features of the original problem. This is the subject to be studied here. Let us refer to Guyon and Elissef (2003) for a comprehensive survey.

It is conventional to distinguish between **wrapper** and **filter** modes of the feature selection process. Wrapper methods contain a well-specified objective function, which should be optimized through the selection. The algorithmic process usually involves several iterations until a target or convergence is achieved. **Feature filtering** is a process of selecting features without referring back to the data classification or any other target function. Hence we find filtering as a more suitable process that may be applied in an **unsupervised** manner.

Unsupervised feature selection algorithms belong to the field of unsupervised learning. These algorithms are quite different from the major bulk of feature selection studies that are based on supervised methods (e.g., Guyon and Elissef, 2003, Liu and Wong, 2002), and compared to the latter are relatively overlooked. Unsupervised studies, unaided by objective functions, may be more difficult to carry out, nevertheless they convey several important theoretical advantages: they are unbiased, by neither the experimental expert nor by the data-analyst, can be preformed well when no prior knowledge is available, and they reduce the risk of overfitting (in contrast to supervised feature selection that may be unable to deal with a new class of data). The downside of the unsupervised approach is that it relies on some mathematical principle, like the one to be suggested in this study, and no guarantee is given that this principle is universally valid for all data. A common practice to resolve this quandary is to demonstrate the success of the method on various biological datasets and compare the results obtained by the method with external knowledge.

Existing methods of unsupervised feature filtering include ranking of features according to range or variance (e.g., Herrero, 2003, Guyon and Elissef, 2003), selection according to highest rank of the first principal component ('Gene shaving' of Hastie *et al*. 2000,

*To whom correspondence should be addressed.

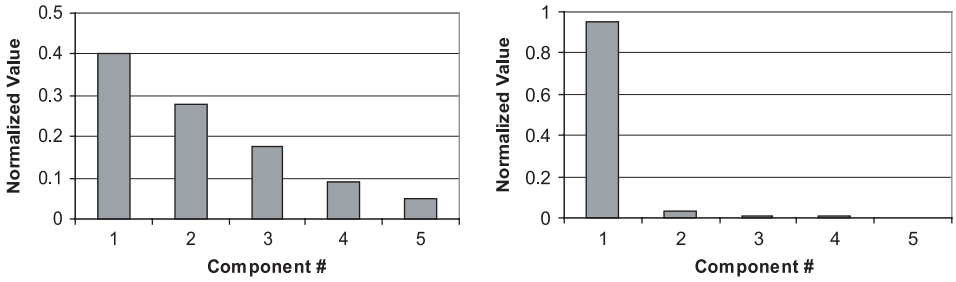


Fig. 1. A comparison of two eigenvalue distributions; the left has high entropy (0.87) and the right one has low entropy (0.14).

Ding 2003) and other statistical criteria. An example of the latter is Ben-Dor *et al*., (2001) where all possible partitions of the data are considered and the corresponding features are labeled. The partitions with statistical significant overabundance are selected. Another example is of Wolf *et al*., (2005), who optimize a function based on the spectral properties of the Laplacian of the features.

Here we present an intuitive, efficient and deterministic principle, leaning on authentic properties of the data, which serves as a reliable criterion for feature ranking. We demonstrate that this principle can be turned into efficient and successful feature selection methods. They compete favorably with other popular methods.

2 METHODS

2.1 Mathematical framework and notations

Let us consider a dataset of n instances[1] $A_{[nXm]} = \{\bar{A}_1, \bar{A}_{2,\ldots,} \bar{A}_{i,\ldots,} \bar{A}_n\}$, where each instance, or observation, $\bar{A}_i$ is a vector of m measurements or features. The objective is to define a subset of features $\tilde{M}$, of size $m_c < m$, that, in a sense to be defined below, best represents the data.

In PCA (or SVD) studies it is conventional to regard the best representation as the minimal least-square approximation of the original matrix (Wall *et al*., 2003). This principle can be followed also in feature extraction but it has the disadvantage that it may preserve too many properties of the data, including systematic noise. We will define our 'best approximation' using a principle based on SVD-entropy, and subject it to an a-posteriori test: given different selection rules of features choose the ones that prove useful as basis for the best fit to labeled data, e.g., perform clustering within the data-space spanned by the selected features and compare the results with known classification. This comparison will be performed using the Jaccard score.

$$J = \frac{n_{11}}{n_{11} + n_{01} + n_{10}} \tag{1}$$

where n_{11} is the number of pairs of instances that are classified together, both in the 'expert' classification and in the classification obtained by the algorithm; n_{10} is the number of pairs that are classified together in the 'expert' classification, but not in the algorithm's classification; n_{01} is the number of pairs that are classified together in the algorithm's classification, but not in the 'expert' classification.

The Jaccard score reflects the 'intersection over union' between the algorithm's clustering assignments and the expected classification. Its values range from 0 (no match) to 1 (perfect match).

2.2 Ranking by SVD-Entropy

Alter *et al*., (2000) have defined an SVD-based entropy of the dataset. Denote by s_j the singular values of the matrix A. s_j^2 are then the eigenvalues of the nxn matrix AA^t. Let us define the normalized relative values (Wall *et al*., 2003): and the resulting

$$V_j = s_j^2 \Big/ \sum_k s_k^2 \tag{2}$$

dataset entropy (Alter *et al*., 2000):

$$E = -\frac{1}{\log(N)} \sum_{j=1}^{N} V_j \log(V_j) \tag{3}$$

This entropy varies between 0 and 1. E = 0 corresponds to an ultra-ordered dataset that can be explained by a single eigenvector (problem of rank 1), and E = 1 stands for a disordered matrix in which the spectrum is uniformly distributed. Figure 1 demonstrates two examples of 5 eigenvalues, one with high entropy (left, 0.87) and the other with low entropy (right, 0.14). As can be seen in Figure 1, when the entropy is very low, one expects a very non-uniform behavior of eigenvalues. One should not confuse the standard definition of entropy, based on probabilities (Shannon, 1948), with the one used here, which is based on the distribution of eigen- (or singular) values. Although standard entropy considerations appear in feature selection methods, such as the supervised bottleneck approach (Tishby *et al*., 2000), the use of SVD-entropy for feature selection is a novel approach.

We define the contribution of the *i-th* feature to the entropy (CE_i) by a leave-one-out comparison according to

$$CE_i = E(A_{[nXm]}) - E(A_{[nX(m-1)]}) \tag{4}$$

where, in the last matrix, the *i-th* feature was removed.

Thus we can sort features by their relative contribution to the entropy. Let us define the average of all CE to be c and their standard deviation to be d. We distinguish then between three groups of features:

(1) $CE_i > c + d$, features with high contribution

(2) $c + d > CE_i > c\text{-}d$ features with average contribution

(3) $CE_i < c\text{-}d$ features with low (usually negative) contribution

Features in the first group (high CE) lead to entropy increase; hence they are assumed to be very relevant to our problem. Retaining these features we expect the instances to be more evenly spread in the truncated SVD space. The features of the second group are neutral. Their presence or absence does not change the entropy of the dataset and hence they can be filtered out without much information loss. The third group includes features that reduce the total SVD-entropy (usually $c\text{-}d < 0$). Such features may be expected to contribute uniformly to the different instances, and may just as well be filtered out from the analysis.

The first feature selection method that we propose is to limit oneself to the first group of features according to the CE ranking. A will then be represented by a new matrix of rank m_c, the number of features in group 1. Several other feature selection methods are suggested in the next section. In all of them we assume that the same value of m_c continues to serve as the right guide for optimal dimensionality reduction.

2.3 Three Feature Selection Methods

Entropy maximization can be implemented in three different ways, as is also the case in other feature selection methods.

[1]In this paper A (or $A_{[nXm]}$) is a matrix and $\overline{A}$ (or $\overline{A_i}$) is a vector.

1. Start with $\tilde{M} = \varnothing$ and $M' = M$
2. Select the element with the **highest** CE. Remove it from M', insert it into $\tilde{M}$
3. While size of $\tilde{M} < m_c$
   a. For each element in M' ($\forall m \in \tilde{M}$) compute its CE score on $\tilde{M}$ ($E(A_{\tilde{M}+i}) - E(A_{\tilde{M}})$)
   b. Select the element with the **highest** CE Score → remove from M', insert into $\tilde{M}$
4. End

Box 1: Pseudo-code of Forward Selection method FS1

1. Start with $\tilde{M} = \varnothing$ and $M' = M$
2. While size of $\tilde{M} < m_c$
   a. Select the element in M' ($\forall m \in \tilde{M}$) with the **highest** CE Score
   b. Remove from M', insert into $\tilde{M}$
3. End

Box 2: Pseudo-code of Forward Selection in method FS2

1. Start with $\tilde{M} = M$ and $M' = \varnothing$
2. While size of $\tilde{M} > m_c$
   a. Select the element in $\tilde{M}$ with the **lowest** CE Score
   b. Remove from $\tilde{M}$, insert into M'
3. End

Box 3: Pseudo-code of Backward Elimination method BE

(1) Simple ranking (SR): select m_c features according to the highest ranking order of their CE values.

(2) Forward Selection (FS): here we consider two implementations.

   (a) FS1: Choose the first feature according to the highest CE. Choose among all other features the one which, together with the first feature, produces a 2-feature set with highest entropy. Continue with iteration over all *m-2* features to choose the third according to maximal entropy, etc, until m_c features are selected (Box 1).

   (b) FS2: Choose the first feature as before. Recalculate the *CE* values of the remaining set of size *m-1* and select the second feature according to the highest *CE* value. Continue the same way until m_c features are selected (Box 2).

(3) Backward Elimination (BE): Eliminate the feature with the lowest CE value. Recalculate the CE values and iteratively eliminate the lowest one until mc features remain (Box 3).

One may view the different methods also as specifying alternative ranking methods. Whereas SR ranks the features according to their original CE values, FS1, FS2 and BE introduce other ranking orders through the algorithms defined above. In the examples studied below we display rankings for the entire range of 1 to *m*.

In an appendix we analyze the computational complexity of all these methods. SR is the fastest one and BE is the most cumbersome one for large numbers of features. In the examples to be discussed next, we will compare the different methods with one another. However, because of complexity, the BE method will be used in only one of the examples.

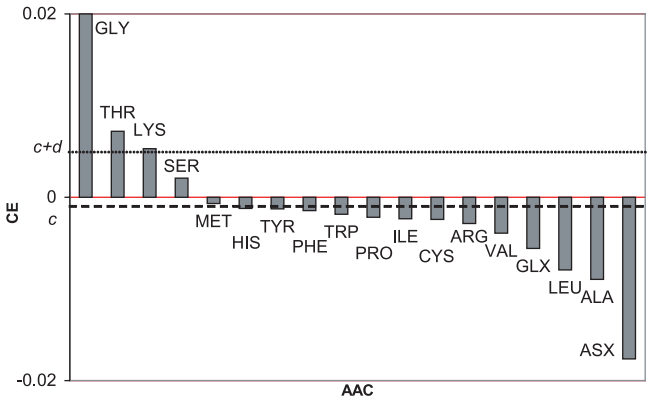


Fig. 2. CE of the 18 Amino Acid Compositions (AAC) of the virus dataset. ASX stands for ASN and ASP and GLX for GLN and GLU. The dashed line represents the value of *c* and the dot-dashed line the value of *c+d*.

3 Results

Our four feature filtering methods were compared with each other and with two known methods: Variance Selection (VS) and Gene Shaving (GS). The latter is a variation of a method of Hastie *et al*. (2000) which removes features iteratively according to their lowest correlations with the first principal component. For comparison we also look at results of random feature selection on several benchmarks.

3.1 The viruses dataset of Fauquet, 1988

This is a dataset of 61 rod-shaped viruses affecting various crops (tobacco, tomato, cucumber and others) originally described by Fauquet *et al*. (1988) and analyzed more thoroughly by Ripley (1996). There are 18 measurements of Amino Acid Compositions (AAC) for the coat proteins of the virus that serve as 18 features. The viruses are known to be classified into four classes: Hordeviruses (3), Tobraviruses (6), Tobamoviruses (39) and Furoviruses (13).

Figure 2 displays the CE values of all 18 features. Our criterion sets $m_c = 3$. We test the performance of the system for the entire *m* range to see if this choice makes sense. Before doing so, let us display the ranking orders of all methods in Table 1. By definition, SR has the same ranking order as CE in Figure 2. In this problem, BE turns out to lead to the same order as FS1, and all our three methods agree with each other on the first three features to be selected. We include in Table 1 also the ranking order of VS (variance selection) and GS (gene shaving). The two last ones are highly correlated with each other (Spearman correlation 0.76) but highly uncorrelated with our three methods (see Supplementary Material for more details). In particular note that VS chooses ASX and GLX as its second and third features, whereas for our three methods these two features are unfavorable (15^{th} to 18^{th}) choices.

Next we evaluate the subset selection using the Jaccard score. This is done by applying the QC clustering algorithm (Horn and Gottlieb, 2002) on the 61 viruses described by the selected subset of features. QC was applied after reduction of each space to normalized 3-space dimensions, using the parameter $\sigma = 0.5$ (for details see Varshavsky *et al*., 2005, and COMPACT[2]). Results are shown in

[2]http://adios.tau.ac.il/compact or http://www.protonet.cs.huji.ac.il/compact

Table 1. Ranking of the 18 Amino Acid Compositions of the virus dataset according to various feature filtering methods. Colors from white to black match the numbers that reflect the ranking of each method

AAC	SR	FS1/BE	FS2	VS	GS
GLY	1	1	1	1	9
THR	2	2	2	6	6
LYS	3	3	3	4	14
SER	4	13	4	5	4
MET	5	4	15	16	17
HIS	6	6	7	15	16
TYR	7	8	13	13	13
PHE	8	7	5	14	11
TRP	9	5	16	17	15
PRO	10	11	6	11	10
ILE	11	10	11	12	12
CYS	12	9	18	18	18
ARG	13	12	10	8	8
VAL	14	14	8	9	7
GLX	15	16	9	3	2
LEU	16	15	14	10	5
ALA	17	17	12	7	3
ASX	18	18	17	2	1

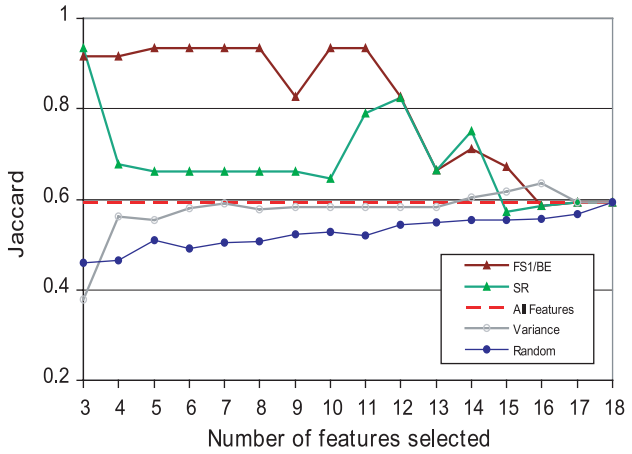


Fig. 3. Filtering quality of the virus dataset is tested by Jaccard scores of clustering performed in spaces spanned by them (see text). Best results are obtained for FS1 (identical with BE in this case) and SR for $m_c = 3$. FS1 continues to perform very well with more features. Feature selection according to VS performs worse. For comparison we include also an evaluation based on a large group of random order rankings.

Figure 3 for three of our four methods. All three do exceedingly well at the three features level (J > 0.9) whereas the variance method obtains J = 0.4. Note that our methods, with our choice of m_c, lead to a much better result than J = 0.6, obtained when all 18 features are taken into account. This exemplifies the importance of keeping features that maximize the entropy. The feature ranking of FS1 and BE is the only one that keeps performing very well with more than three selected features. Similar relative successes of feature selection evaluation (although less favorable J-scores) were obtained with other clustering methods, such as K-means. This comparison, as well as other details that could

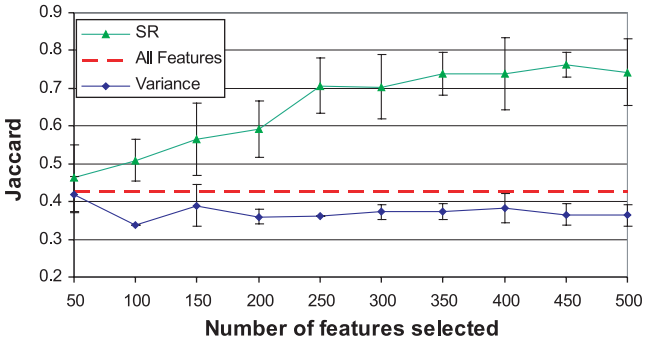


Fig. 4. Clustering quality of two feature selection methods. Results are averages of 100 runs of K-Means clustering.

not be fitted into this paper, can be found in the Supplementary Material[3].

Fauquet *et al*. (1987) have argued that the AAC of the coat protein of plant viruses are specific to the structure of the viral particle, to the mode of transmission and to sub-grouping of viruses to distinctive classes. Our results indicate that choosing only 3–4 features correctly, not only preserves the classification but allows much better performance with minimal failure. It is interesting to note that the 3 highest-ranking amino acids, GLY, THR and LYS are not dominating the coat proteins. These amino acids account for only 13–21.5% of the coat proteins, a fraction that is similar to the average percentage in the entire proteins database (18.3%). Further investigation shows that neither their size nor polarity or electric charges differentiate these three amino acids from the remaining. Nevertheless, since GLY, THR, LYS and MET (the fourth ranked AAC, according to the FS1 method) represent different functional groups, we conclude that the FS1/BE ranking is consistent with selecting amino acids that carry different physico-chemical properties.

3.2 The MLL dataset of Armstrong *et al.*, 2002

The second dataset that we apply our methods to is that of Armstrong *et al*., 2002, who have attempted to cluster data of three Leukemia classes: lymphoblastic Leukemia with MLL translocations and conventional acute lymphoblastic (ALL) and acute myelogenous Leukemias (AML). In the experiment, 12582 gene expressions were recorded, using Affymetrix U95A chips on 72 patients, 20 of which diagnosed as MLL, 24 ALL and 28 AML. They showed that these 3 Leukemia types can be divided according to some gene expression. However, when filtering in an unsupervised manner (selecting 8700 genes that show some variability in expression level), the clustering results were unsatisfactory and much inferior to a supervised selection of 500 genes that best separate between the cancer patients.

Applying our CE criteria we use the method SR, and compare clustering of these feature-filtered data with VS (Figure 4). Clustering was performed by K-Means, averaging over 100 runs and using K = 3 with data projected onto a unit sphere in 3D-reduced space (Varshavsky *et al*., 2005). The asymptotic Jaccard score is J = 0.426 for this K-Means method. As can be seen in Figure 4 VS provides no improved quality, whereas SR leads to J-values

[3]http://adios.tau.ac.il/compact/UFF/SUPP

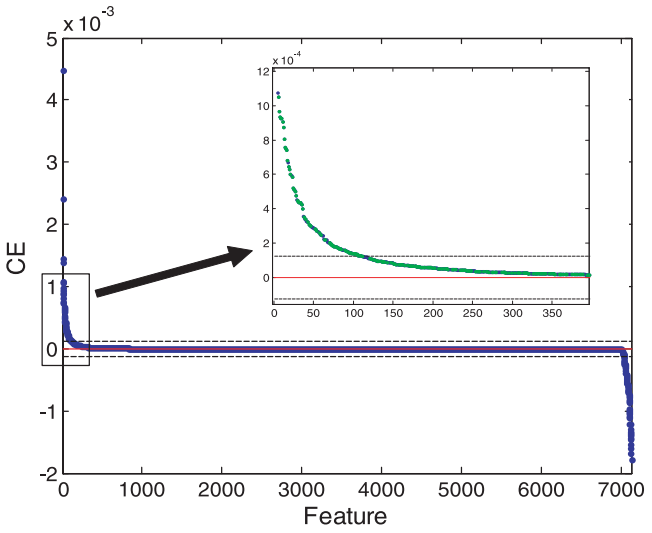


Fig. 5. CE of the 7129 genes of the Golub dataset ($c = 0$, dashed lines represent $c \pm d$). The inset zooms into the highest-ranked 300 genes, with bright dots signifying the top 100 features according to the FS1 method

between 0.7 and 0.8 for filtered gene groups of sizes 250 to 450. The preferred m_c value according to $c + d$ of SR is 254. Better results can be obtained by using the QC algorithm, but the same trend and conclusions regarding feature selection hold also there. It is interesting to note that QC clustering of our unsupervised SR method, for $m_c = 254$, reaches J = 0.85 (see supplementary).

We display the K-Means analysis in Figure 4, in spite of its poorer performance compared to QC, in order to emphasize that the quality of the feature filtering method is independent of the clustering-test performed on the filtered data.

3.3 The Leukemia dataset of Golub *et al.*, 1999

After demonstrating the effectiveness of our methods on both small and large datasets, we choose a third dataset (Golub *et al.*, 1999) that has served as a benchmark for several clustering algorithms (Sharan and Shamir, 2000, Getz *et al.*, 2000 and more) and feature selection methods (e.g., Liu B. *et al.*, 2004, Liu H. *et al.*, 2002). The experiment sampled 72 Leukemia patients with two types of Leukemia, ALL and AML. The ALL set is further divided into T-cell Leukemia and B-cell Leukemia and the AML set is divided into patients who have undergone treatment and those who did not. For each patient, an Affymetrix GeneChip measured the expression of 7129 genes. The task is clustering into the four correct groups within the 72 patients in a [7129x72] gene-expression matrix. This clustering task is quite difficult. Using the QC method (in normalized 5 dimensions with $\sigma = 0.54$), applied to the data without feature selection, one obtains J = 0.707, which is the best score for a variety of clustering algorithms (Varshavsky *et al.*, 2005).

The CE values for the 7129 features of this problem are displayed in Figure 5. Most of the features have a zero score. There are about 150 large CE values (see Figure 5) and about the same number of small CE values. The bright color within the inset indicates the first 100 features selected by FS1. While their ordering is different from the SR ranking, most of them belong, as expected, to the class of large CE values. The overlaps of the first leading features of SR

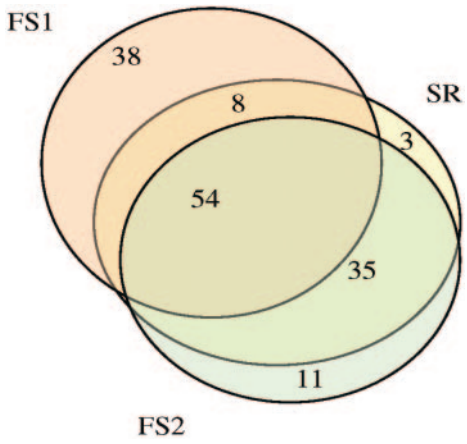


Fig. 6. Venn diagram of relations among the first 100 features selected by different methods.

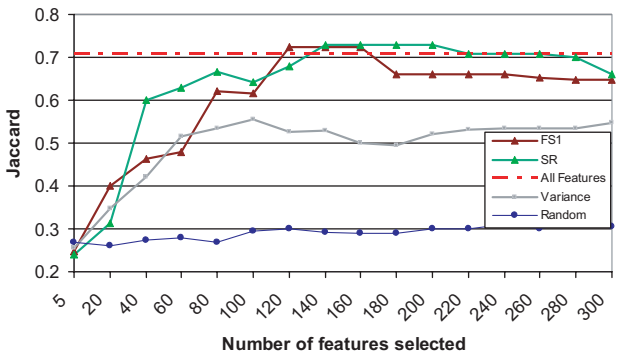


Fig. 7. Jaccard scores of QC clustering for different feature filtering methods on small gene subsets of the Golub data.

with those of FS1 and FS2 are shown in the Venn diagrams of Figure 6.

Next we turn to testing the filtering methods to see how well they do in the clustering task, i.e. what are the Jaccard scores that are obtained by applying an identical clustering algorithm to the different spaces spanned by the selected features. The clustering algorithm is the QC method mentioned above. Figure 7 shows that good results can be obtained by our filtering methods once the gene subset is larger than 100 or so. For feature sets of sizes 120 to 200 we find selections (of FS1 and SR) that lead to Jaccard scores that are better than J = 0.707, the asymptotic limit. Gene subsets larger than 300 result in Jaccard scores below the asymptotic limit (for a complete list, see supplementary). Also in this problem the GS results are inferior to those of the other methods.

3.3.1 Biological interpretations of the Leukemia dataset of Golub et al., 1999 It is clearly of interest to look at the 100 or so genes that participate in the sections that lead to the best Jaccard score. In Figure 6 we saw that there exists a substantial overlap between the choices of our three different methods. To study the biological significance of our subset of overlapping 54 genes we have run a GO enrichment analysis (NetAffx™ web tool[4]) on this subset. As

[4]http://www.affymetrix.com/analysis/index.affx

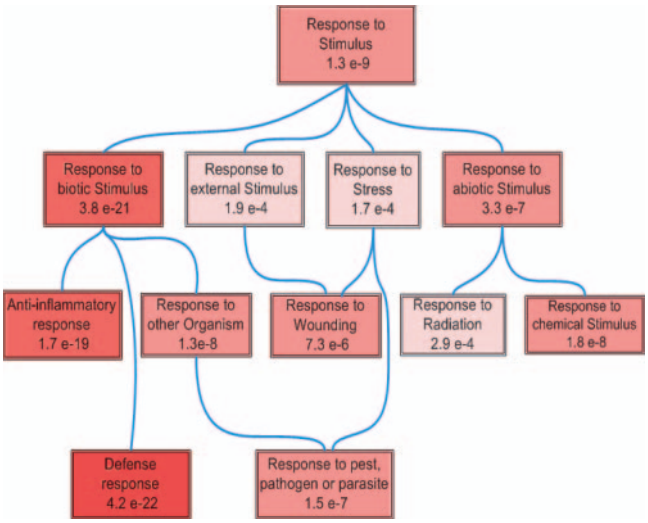


Fig. 8. Diacyclic graph of GO enrichment. Shown are GO nodes (Camon *et al*, 2004) with significant p-value of enrichment as determined by the NetAffx™ tool[4] (p-value < 5e-4). The color of each node matches its significance level (along the spectrum of red shades, light: lowest to dark: highest).

displayed in Figure 8 (and supplementary), we are able to assign some prevalent biological processes to the selected genes.

The association of our selected 54 genes with functional annotation related to defense, inflammation and response to pathogen (with p-value ranging from e-7 to e-22) is intriguing (Figure 8). It may underlie the difference in AML and ALL in view of the different susceptibility of the patients to treatment such as chemo and radiotherapy. Thus the listed protein processes may not only be considered as 'subtype cancer markers' but as an indication of the biological properties of the cancerous cells. Specifically, cellular response to pathogen, to stress and to inflammation may be different for AML and ALL. It may also provide a focused hypothesis towards the processes and mechanisms that can be used as a follow up in monitoring the outcome of therapy in case of Lymphoma.

4 Discussion

We have introduced a novel principle for unsupervised feature filtering that is based on maximization of SVD-entropy. The features can be ranked according to their CE-values. We have proposed four methods based on this principle and have tested their usefulness on three different biological benchmarks. Our methods outperform other conventional unsupervised filtering methods. This is clearly brought out by the examples that we have analyzed. More details are provided by our Supplementary Material[5]. In particular, it is striking to note how much more successful our methods are compared to VS, the popular variance ordered method.

The major theoretical difference between the two approaches is that VS relies on a measurement of one feature at a time. The entropy-based approach, as implemented by the CE calculation, takes into account the interplay of all features. In other words, the contribution of a feature, its CE, depends on the behavior of all other features in the problem. Thus variance is only one of the factors that affect the CE value. The CE value depends also on the correlations (or the absence thereof) of a given feature with all others. The difference between the ranking of SR and VS in Table 1 bears evidence to the difference between the two methods.

We have demonstrated that our selected features have important biological significance, through a GO enrichment analysis of the genes in the Golub dataset. A similar analysis of the Armstrong dataset is presented in the Supplementary Material[5]. In the virus dataset, we have shown that the FS1/BE filtering method works exceedingly well for a large range of numbers of features. The biological significance of the relevant choices of amino-acids remains to be uncovered.

The CE ranking leads to an estimate of the optimal m_c choice. This is an important point by itself. In other methods, such as VS, it is almost impossible to make this choice on the basis of variation of feature properties. Conventionally one makes therefore an arbitrary choice, such as selecting 10% or 50% of the features. In the three datasets discussed in our paper it seems quite clear that our suggested optimal m_c, as judged from the CE scores, leads indeed to optimal results. The improved Jaccard scores indicate that the selected m_c features have biological significance.

Our four methods differ in computational complexity. SR is the simplest one, since it relies just on sorting the initial CE values. In an appendix we compare its complexity with that of the other methods. The relative values depend on the choice of m_c (the size of the subset).

FS1 chooses features that lie high on the original CE-score, hence its optimal selected set will have a large intersection with that of SR. Nonetheless, for small numbers of selected features, the order may be very important. Thus, in the virus problem, FS1 turns out to be much more successful than SR. In the Leukemia datasets, where reasonable results were obtained for larger feature sets, FS1 was not found to be significantly better than SR. Biologically one may expect the appearance of features that are degenerate with one another, i.e. have quite identical behavior on all instances. Such duplicity can be included by the SR method but excluded by the FS1 one.

Our optimal feature-filtered sets in the two Leukemia problems turn out to include just few percents of all genes. Thus a CE-analysis indicates that a small subgroup of all genes is the most relevant one to the data in question. We have seen that this relevance is borne out by both Jaccard scores and GO enrichment analysis. The pursuit of small feature sets is often guided by wishful thinking that the essence of biological importance can be reduced to a small causal set. Here we find that the small number obtained in our analysis is an emerging phenomenon, and may be regarded as a true biological result.

ACKNOWLEDGEMENTS

We thank Alon Kaufman and Nati Linial for stimulating discussions and suggestions, and Orly Alter for technical and theoretical assistance. R.V. is supported by SCCB, the Sudarsky Center for Computational Biology in the Hebrew University of Jerusalem. This study was supported by the EU Framework VI NoE

[5]http://adios.tau.ac.il/compact/UFF/SUPP

DIAMONDS consortium, and also partially supported by the Israel Science Foundation (grant No. 39/02).

REFERENCES

Alter, O., Brown, P.O. and Botstein, D. (2000) Singular value decomposition for genome-wide expression data processing and modeling, PNAS, 97, 10101–10106.

Armstrong, S.A., Staunton, J.E., Silverman, L.B., Pieters, R., den Boer, M.L., Minden, M.D., Sallan, S.E., Lander, E.S., Golub, T.R. and Korsmeyer, S.J. (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia, 30, 41–47.

Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, LAPACK User's Guide (http://www.netlib.org/lapack/lug/lapack_lug.html), Third Edition, SIAM, Philadelphia, 1999.

Ben-Dor, A., Friedman, N. and Yakhini, Z. (2001) Class discovery in gene expression data. RECOMB. 31–38.

Camon E, Barrell D, Lee V, Dimmer E. and Apweiler R. (2003) Gene Ontology Annotation Database—An integrated resource of GO annotations to UniProt Knowledgebase. In Silico Biol., 4: 0002.

Ding, C., He, X., Zha, H. and Simon, H. (2002) Adaptive dimension reduction for clustering high dimensional data. IEEE International Conference on Data Mining. 107–114.

Ding, C.H.Q. (2003) Unsupervised Feature Selection Via Two-way Ordering in Gene Expression Analysis, Bioinformatics, 19, 1259–1266.

Fauquet, C., Desbois, D., Fargette, D. and Vidal, G. (1988) Classification of furoviruses based on the amino acid composition of their coat proteins. In Cooper, J.I. and Asher, M.J.C. (eds), Viruses with Fungal Vectors. Association of Applied Biologists, Edinburgh, 19–38.

Fauquet, C., Thouvenel, J. C. (1987). Viral diseases of plants in Ivory Cost. Intuition et Documentation Technique, 46. ORSTOM, Paris, 243 pp.

Getz, G., Levine, E. and Domany, E. (2000) Coupled two-way clustering analysis of gene microarray data, PNAS, 97, 12079–12084.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. and Lander, E.S. (1999) Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, Science, 286, 531–537.

Guyon, I. and Elisseeff, A. (2003) An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, 3, 1157–1182.

Hastie, T., Tibshirani, R., Eisen, M., Alizadeh, A., Levy, R., Staudt, L., Chan, W., Botstein, D. and Brown, P. (2000) 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns, Genome Biology, 1.

Herrero, J., Diaz-Uriarte, R. and Dopazo, J. (2003) Gene expression data preprocessing, Bioinformatics, 19, 655–656.

Horn, D. and Axel, I. (2003) Novel clustering algorithm for microarray expression data in a truncated SVD space, Bioinformatics, 19, 1110–1115.

Horn, D. and Gottlieb, A. (2002) Algorithm for data clustering in pattern recognition problems based on quantum mechanics, Physical Review Letters, 88.

Liu, B., Cui, Q., Jiang, T. and Ma, S. (2004) A combinational feature selection and ensemble neural network method for classification of gene expression data, BMC Bioinformatics,5. 136

Liu, H., Li, J. and Wong, L. (2002) A Comparative Study on Feature Selection and Classification Methods Using Gene Expression Profiles and Proteomic Patterns. In R. Lathrop, K.N., S. Miyano, T. Takagi, and M. Kanehisa (ed), 13th International Conference on Genome Informatics. Universal Academy Press, Tokyo Japan, 51–60.

Ripley, B.D. (1996) Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge.

Shannon, C. (1948) A mathematical theory of communication,, The Bell system technical journal, 27, 379–423, 623–656.

Sharan, R. and Shamir, R. (2000) CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis. AAAI Press, Menlo Park, CA, 307--316.

Sondberg-Madsen, N., Thomsen, C. and Pena, J.M. (2003) Unsupervised Feature Subset Selection. Workshop on Probabilistic Graphical Models for Classification. 71–82.

Tishby, N., Pereira, F., C. and Bialek, W. (2000) The information bottleneck method, CoRR, physics/0004057

Varshavsky, R., Linial, M. and Horn, D. (2005) COMPACT: A Comparative Package for Clustering Assessment. Lecture Notes in Computer Science. Volume 3759, 159–167. Springer-Verlag.

Wall, M., Rechtsteiner, A. and Rocha, L. (2003) Singular Value Decomposition and Principal Component Analysis. In Berrar, D., Dubitzky, W. and Granzow, M. (eds), A Practical Approach to Microarray Data Analysis. Kluwer, 91–109.

Wolf, L. and Shashua, A. (2005) Feature Selection for Unsupervised and Supervised Inference: The Emergence of Sparsity in a Weight-Based Approach, Journal of Machine Learning Research, 6, 1855--1887.

Xing, E.P. and Karp, R.M. (2001) CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. Bioinformatics, 17, S306–315.

APPENDIX

Computational complexity of the four methods

In the following calculations, we will assume that $m_c < n$, which will give upper bound to the complexity. We will not assume that $m < n$.

The computation of all eigenvalues for a dense symmetric matrix requires $O(p^3)$ operations, where p is the size of the matrix (Anderson, 1999).

We will define the complexity of the initial computation of all CEs to be $O(m^*min(n,m)^3) \equiv K$.

- SR: The computational complexity is lowest for the SR method. There's only one calculation of all CEs, followed by sorting. Hence the complexity is $O(K + m^*logm)$.
- FS1: Calculation of all CEs followed by $(m_c - 1)$ repetitive diagonalization of a growing matrix (from 2 to $(m_c - 1)$), leading to $O(K + m_* m_c^4)$.
- FS2: Calculation of all CEs followed by $(m_c - 1)$ repetitive diagonalization of a decreasing matrix (from m-2 to $(m\text{-}m_c)$), leading to $O(m^5\text{-}(m\text{-}m_c)^5)$. Note that here, if $n < (m\text{-}m_c)$, the complexity is $O(mm_c n^3)$
- BE: Calculation of all CEs followed by $(m\text{-}m_c\text{-}1)$ repetitive diagonalization of a decreasing matrix (from m-2 to $(m_c\text{-}1)$), leading to $O(m^5\text{-}m_c^5)$. Note that here, if n < m, the complexity is reduced to $O((m^2\text{-}m_c^2)n^3)$.

Clearly computational complexity is lowest for the SR method, since only one calculation of all CEs is needed. BE or FS2 have the highest complexity, depending on whether $m > 2m_c$ or not.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e514–e522
doi:10.1093/bioinformatics/btl262

Constructing Near-Perfect Phylogenies with multiple homoplasy events

Ravi Vijaya Satya[1], Amar Mukherjee[1], Gabriela Alexe[2], Laxmi Parida[2] and Gyan Bhanot[2]

[1]School of EECS, University of Central Florida, Orlando FL 32816-2362 USA and [2]Computational Biology Center, IBM T.J. Watson Research Center, York Town Hts. NY 10598 USA

ABSTRACT

Motivation: We explore the problem of constructing near-perfect phylogenies on bi-allelic haplotypes, where the deviation from perfect phylogeny is entirely due to homoplasy events. We present polynomial-time algorithms for restricted versions of the problem. We show that these algorithms can be extended to genotype data, in which case the problem is called the near-perfect phylogeny haplotyping (NPPH) problem. We present a near-optimal algorithm for the H1-NPPH problem, which is to determine if a given set of genotypes admit a phylogeny with a single homoplasy event. The time-complexity of our algorithm for the H1-NPPH problem is $O(m^2(n+m))$, where n is the number of genotypes and m is the number of SNP sites. This is a significant improvement over the earlier $O(n^4)$ algorithm.

We also introduce generalized versions of the problem. The H(1, q)-NPPH problem is to determine if a given set of genotypes admit a phylogeny with q homoplasy events, so that all the homoplasy events occur in a single site. We present an $O(m^{q+1}(n+m))$ algorithm for the H(1,q)-NPPH problem.

Results: We present results on simulated data, which demonstrate that the accuracy of our algorithm for the H1-NPPH problem is comparable to that of the existing methods, while being orders of magnitude faster.

Availability: The implementation of our algorithm for the H1-NPPH problem is available upon request.

Contact: rvijaya@cs.ucf.edu

1 INTRODUCTION

Though the genomic sequence is mostly similar from individual to individual, each individual differs from others in some locations. Studying these variations will help in understanding, diagnosis, and treatment of many genetically inherited diseases. Single Nucleotide Polymorphisms (SNPs) are the most common genetic variations observed. SNPs are loci in the human genome where multiple variants exist at a high enough frequency (>0.05) that the position can be considered polymorphic within the population. Each individual variant in a SNP location is called an allele. It is estimated [HapMapConsortium, 2003] that there are as many as 10 million SNPs in the human genome, which translates to a density of one SNP every three hundred base pairs of DNA. More than 99% of the SNPs in the human genome are bi-allelic.

*To whom correspondence should be addressed.

The human genome is diploid, meaning that in each cell there are two copies of each chromosome. Due to the bi-parental nature of heredity in diploid organisms, one of these copies is derived from the mother and the other is derived from the father. Each of these copies is called a *haplotype*. As we are interested in only the SNP locations in the genome, a haplotype that covers a region of the chromosome with m SNPs is generally represented as a binary vector of length m. The values 0 and 1 represent the two alleles of each SNP. A *genotype* gives combined information about the two haplotypes, and is represented by a length-m vector over the alphabet $\{0, 1, 2\}$. In a genotype g, if $g[i]$ is 0 or 1, it implies that the two haplotypes (h, h') for g are homozygous in the ith SNP with the 0-allele or the 1-allele, respectively. If $g[i] = 2$, it implies that the ith SNP is heterozygous in g. i.e., either $h[i] = 0$ and $h'[i] = 1$, or $h[i] = 1$ and $h'[i] = 0$.

With the current technology, the cost associated with empirically collecting haplotype data is prohibitively expensive. Therefore, only the un-ordered bi-allelic genotype data is collected through empirical means. This necessitates computational techniques for inferring haplotypes from genotypes. Given n genotypes over m SNP sites, the *haplotype inference* (HI) problem is to find a pair of haplotypes for each genotype, so that combining the two haplotypes results in the genotype. This problem is also referred to as the *phase* problem in genotyping. For each genotype, we want to find the most likely pair of haplotypes that might have combined to form the genotype. The haplotype inference problem was first introduced by Clark (1990). Subsequently, multiple formulations were introduced, with different definitions for the optimum solution. Most formulations are based on parsimony, perfect phylogeny or maximum likelihood. A comprehensive survey of the many different variations of the HI problem is provided by Bonizzoni *et al*. (2003).

1.1 Perfect phylogeny

Under the coalescent model of evolution, all the individuals in a population have a common ancestor. Applying the standard infinite sites assumption to the coalescent model leads to the perfect phylogeny model of evolution, which assumes that each site can mutate only once. A perfect phylogeny T for n haplotypes over m SNPs is a tree in which each of the m SNPs labels exactly one edge in T. Each vertex in T is labeled by a haplotype vector. Each of the n haplotypes must label some vertex in the tree.

Applying the coalescent model to the Haplotype Inference problem, Gusfield (2002) introduced a perfect phylogeny formulation

of the problem, called the PPH(Perfect Phylogeny Haplotyping) problem. The perfect phylogeny formulation requires that all the haplotypes that resolve the given set of genotypes describe a perfect phylogeny. The perfect phylogeny model is justified by the block structure of the human genome and the validity of the infinite sites assumption.

Gusfield *et al*. (2002) presented an $O(nm^2)$ algorithm for the PPH problem by reduction to the graph realization problem. Bafna *et al*. (2002) presented a direct solution that takes $O(nm\alpha(nm))$ time. Recently, three independent $O(nm)$ algorithms (Liu and Zhang, 2004; Ding *et al*., 2005; Vijaya Satya and Mukherjee, 2005, 2006) have been developed for the PPH problem.

1.2. Imperfect phylogeny

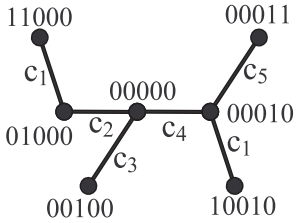
Biological data rarely, if ever, conforms to perfect phylogeny because of repeated mutations and recombinations. However, the deviations from perfect phylogeny are expected to be small within a 'block' of the human genome. When the deviations from perfect phylogeny are small, the phylogenies can be referred to as near-perfect phylogenies. The term 'homoplasy event' is used to refer to a repeated/back mutation. The problem of constructing near perfect phylogenies with multiple homoplasy events has been tackled before (Fernandez-Baca and Lagergren, 2003). The complexity of their algorithm for constructing near perfect phylogenies on a set of n haploid taxa is given by $O(nm^q 2^{q^2 r^2})$, where r is maximum number of alleles in any site, and q is the number of repeated/back mutations. In this paper, we are only concerned with bi-allelic SNP data, and hence $r = 2$. Even in case of bi-allelic data, the above algorithm is clearly impractical for values of q as small as four. Recently Sridhar *et al*. (2005) proposed a more practical algorithm for binary data with complexity $(q+p)^{O(q)}nm + O(nm^2)$ where p is the number of characters that share four gametes with some other character.

In this paper, we deal with restricted versions of the near-perfect phylogeny problem on both haplotype and genotype data and present polynomial time algorithms for these problems. Song *et al*. (2005) have introduced a restricted version of the near-perfect phylogeny haplotyping problem that allows a single homoplasy event. They specifically defined the problem on genotype data and called the problem the H1-Imperfect Phylogeny Haplotyping (H1-IPPH) problem. The notation 'H1' indicates that there is a single homoplasy event in the phylogeny. The acronym IPPH has previously been used (Halperin and Karp, 2004; Kimmel and Shamir, 2005) to refer to the Incomplete Perfect Phylogeny Haplotyping problem. Therefore, in this paper, we rename the problem as the H1- Near-Perfect Phylogeny Haplotyping problem (H1-NPPH). Song *et al*. (2005) first identify the column with the homoplasy event, construct a perfect phylogeny T' for the remaining columns, and then convert T' into an H1-NPP T that includes the column with the homoplasy event. In converting T' into T, the procedure followed by Song *et al*. (2005) is to remove pairs of edges from T' and carry out certain tests on the disconnected subtrees produced as a result of removing the pair of edges from T'. The overall complexity of the algorithm is $O(n^4)$.

Our fundamental approach is similar to that presented in Song *et al*. (2005). However, we observe that removing pairs of vertices from T' leads to a faster algorithm than removing pairs of edges from T'. This observation leads to a faster $O(m^2(n+m))$ algorithm that can be easily extended to handle multiple homoplasy events. Based on this observation, we present a generalized framework for constructing near-perfect phylogenies(NPPs) that involve multiple homoplasy events, both for haplotype and genotype data. We define an H(1, q) NPP as a near-perfect phylogeny involving q homoplasy events in a single site. Similarly, a H(p, q) -NPP is a near perfect phylogeny in which at most p sites have homoplasy events, with at most q homoplasy events in each site. Under this notation, a near-perfect phylogeny with a single homoplasy is denoted as the H(1, 1)-NPP.

	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	1	0	0	0
r_3	0	0	1	0	0
r_4	1	0	0	1	0
r_5	0	0	0	1	1

(a)



(b)

Fig. 1. (a) A haplotype matrix M; (b) A phylogeny T for M.

In Section 2.1, we present polynomial-time algorithms for constructing near-perfect phylogenies for haplotype data. In Section 2.2, we extend these algorithms to deal with genotype data. Testing an implementation of our H1-NPPH algorithm on simulated data, we show that our algorithm is extremely fast while having comparable accuracy to that of the popular PHASE (Stephens *et al*., 2001) program.

2. METHODS AND ALGORITHMS

2.1 Constructing Near-Perfect Phylogenies from haplotype data

In the following, we present polynomial-time algorithms for restricted versions of Near-Perfect Phylogeny (NPP) problem. In all the problems that we describe in this section, the input is an $n \times m$ matrix M over the alphabet $\{0, 1\}$, where the columns $c_1, c_2, \ldots, c_m$ indicate sites and the rows $r_1, r_2, \ldots, r_n$ indicate samples. Given that the matrix M does not admit a perfect phylogeny, we want to construct a near-perfect phylogeny for M that is the closest to a perfect phylogeny. We use the terms 'column' and 'site' interchangeably in the rest of this paper.

Throughout this paper, we assume that the deviations from perfect phylogeny are only due to violations of the infinite sites assumption—i.e, due to recurrent or back mutations. The algorithms we present construct un-rooted phylogenies. There is no distinction between a recurrent mutation and a back mutation in an un-rooted phylogeny.

We define the following terms. An ordered pair of values *(a,b)*, $a \in \{0,1\}$, $b \in \{0,1\}$, is said to be *induced* by a pair of ordered columns (i,j) if there is a row r in M such that $M[r,i] = a$ and $M[r,j] = b$. The set of ordered pairs induced by a pair of columns (i,j) is denoted by $I(i,j)$. According to the well-established four-gamete test [11], the matrix M does not admit a perfect phylogeny if $|I(i,j)| = 4$ for any pair of columns (i,j). We say that two columns i and j *conflict* with each other if $|I(i,j)| = 4$. A *conflict graph* $G_c = (V,E)$ is a graph in which each vertex $v_i \in V$ corresponds to a column c_i in M. An edge (v_i, v_j) is in E if the sites c_i and c_j conflict with each other.

The general definition of a phylogeny is that the phylogeny is a tree in which the leaves represent the input taxa. In this paper, we are constructing character-based phylogenies, and hence we are only interested in the topology of the phylogeny. Therefore we use the term phylogeny to refer to an edge and vertex labeled tree T. Each edge in T is labeled by a site in M, and indicates a mutation in that site. An example of a phylogeny is shown in Figure 1. Each vertex in the phylogeny is labeled by a 0-1 vector of length m,

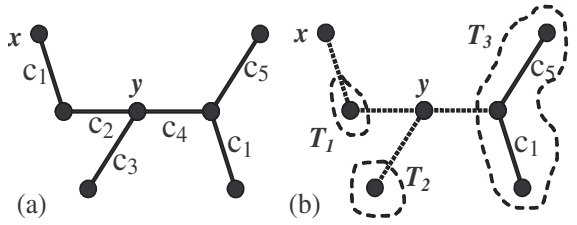


Fig. 2. (a) The tree T before removing the vertices x and y; (b) The three connected components T_1, T_2 and T_3 after removing the vertices x and y.

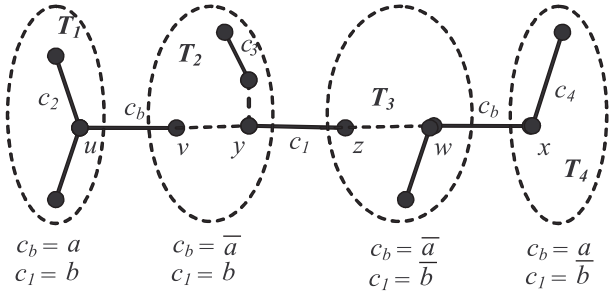


Fig. 3. Illustration of Theorem 1.

and indicates the state of each site at the vertex. For any vertex v, we denote the vertex label of v as $\mathcal{L}(v)$. Since T is a phylogeny for M, for each row r in M, there must be a vertex v such that $\mathcal{L}(v) = M[r]$. This mapping of a row r to a vertex v is represented using the notation $v(r) = v$. Multiple rows in M might map to the same vertex in T, and some vertices in T might not represent any row in M. Notice that the phylogeny in Figure 1 is not a perfect phylogeny. There are two edges in T labeled with column c_1.

Removing a set of vertices S_c from any tree T divides T into a set of connected (trivial or non-trivial) components denoted by $T_{/S_c}$. Note that, since T is a tree, each connected component $T_i \in T_{/S_c}$ will also be a tree. For any connected component T_i of T, we define $R(T_i)$ as the set of rows of M that map to any vertex in T_i. A column c is said to be *non-polymorphic* in T_i if the column c has the same state in each row $r \in R(T_i)$. For example, refer to Figure 2a, which is the same phylogeny as in Figure 1. The three connected components produced by removing the vertices x and y in Figure 2a are shown in Figure 2b (in dotted regions). In the matrix M, the row r_2 maps to T_1, r_3 maps to T_2, and the set of rows $\{r_4, r_5\}$ map to T_3. All the columns are non-polymorphic in T_1 and T_2. However, columns c_5 and c_1 are *polymorphic* in T_3. Columns c_2, c_3 and c_4 are non-polymorphic in T_3.

2.1.1 The H1-NPP construction problem In the following, we describe the conditions under which a given set of haplotypes admit an H1-NPP. There are efficient algorithms to determine if the matrix M admits a perfect phylogeny. When M does not admit a perfect phylogeny, the problem is to construct an H1-NPP for the matrix M, or determine that M does not admit an H1-NPP. For simplicity, we call the H1-NPP construction problem as the H1-NPP problem in the rest of the paper.

Let M be a matrix that does not admit a perfect phylogeny, but admits an H1-NPP. Let c_b be the column with the recurrent mutation. Let T be the H1-NPP for M. By definition, if an edge (u, v) is labeled by a site i, it implies that $\mathcal{L}(u)[i] = \overline{\mathcal{L}(v)[i]}$. Clearly, there will be two edges in T that are labeled with c_b. Let the two edges be (u, v) and (w, x), as shown in Figure 3. We call the path between the two vertices v and w as the *recurrent mutation path*, or *RMP*. Let S be the set of all sites, i.e., $S = \{c_1, c_2, \ldots, c_m\}$. Let S_{RMP} be the set of sites that label an edge in *RMP*. Let S_e be the set of sites other than c_b that are not in RMP. i.e., $S_e = S - \{S_{RMP} \cup \{c_b\}\}$.

THEOREM 1. *Every site $c \in S_{RMP}$ conflicts with c_b, and every site $c \in S_e$ does not conflict with c_b.*

PROOF. Let $\mathcal{L}(u)[c_b] = a$. Clearly, $\mathcal{L}(v)[c_b] = \bar{a} = \mathcal{L}(w)[c_b]$ and $\mathcal{L}(x)[c_b] = a$. For any site $c \in S_{RMP}$, $\mathcal{L}(v)[c] = \overline{\mathcal{L}(w)[c]}$. The site c_1 connecting the vertices y and z in Figure 3 is such a site. Let $\mathcal{L}(y)[c_1] = b$, which implies that $\mathcal{L}(z)[c_1] = \bar{b}$. The phylogeny T can be divided into four subtrees T_1, T_2, T_3 and T_4 with respect to the sites c_b and c_1, as shown in Figure 3. The pair of sites (c_b, c_1) take the states (a,b), $(\bar{a}, b)$, $(\bar{a}, \bar{b})$ and $(a, \bar{b})$, in subtrees T_1, T_2, T_3 and T_4, respectively. Now, $R(T_1), R(T_2), R(T_3)$ and $R(T_4)$ are all non-empty. This is because the matrix M will admit a perfect phylogeny if $R(T_1)$ or $R(T_4)$ are empty, and c_1 need not be in RMP if $R(T_2)$ or $R(T_3)$ are empty. Therefore, $|I(c_b, c_1)| = 4$, and hence c_b conflicts with c_1.

It can similarly be shown that every site $c \in S_e$ will not conflict with c_b. Sites c_2, c_3 and c_4 in Figure 3 are examples of such sites. $\diamond$

As explained before, $T_{/\{u,v,w,x\}}$ is the set of connected components generated by removing vertices u, v, w and x from T. Removing the vertices u, v, w and x removes both the edges labeled with c_b from T. Therefore, no connected component in $T_{/\{u,v,w,x\}}$ will have an edge labeled with c_b. Therefore, the column c_b will be non-polymorphic within any connected component $T_i \in T_{/\{u,v,w,x\}}$.

We will now state and prove a theorem that gives the necessary and sufficient conditions for a haplotype matrix to admit a H1-NPP. Let M be a matrix such that M does not admit a perfect phylogeny, but the matrix M' produced by removing a column c_b from M admits a perfect phylogeny T'. Since the rows in M correspond one-to-one with rows in M', the rows in M can be mapped to vertices in T'. It will be helpful to visualize the matrix M as the matrix M' with a single column c_b appended as the rightmost column of M. We state the following theorem:

THEOREM 2. *The matrix M admits an H1-NPP iff there are two vertices x and y in T' such that the site c_b is non-polymorphic in every connected component in $T'_{/\{x,y\}}$.*

PROOF. Let $T'_{/\{x,y\}} = \{T_1, T_2, \ldots . T_k\}$, as shown in Figure 4a, where $k = d(x) + d(y) - 1$, $d(x)$ is the degree of x and $d(y)$ is the degree of y in T'. We show that we can construct an H1-NPP T for M by expanding the vertices x and y into edges labeled with c_b. We start with an empty tree T. We replace x with two new vertices x_0, x_1, and y with two new vertices y_0 and y_1, and add two edges (x_0, x_1) and (y_0, y_1), both labeled with c_b. The two vertices x_0 and x_1 are labeled based on the label of the vertex x in T' as—$\mathcal{L}(x_0)[i] = \mathcal{L}(x_1)[i] = \mathcal{L}(x)[i]$ for every column $i \neq c_b$. This is equivalent to taking the matrix M' and associating the vertex label of x in T' to both the vertices x_0 and x_1. The site c_b is now associated with the edge (x_0, x_1) as follows: $\mathcal{L}(x_0)[c_b] = 0$, and $\mathcal{L}(x_1)[c_b] = 1$. The vertices y_0 and y_1 are similarly labeled based on the label of the vertex y in T' in every site other than c_b. In site c_b, $\mathcal{L}(y_0)[c_b] = 0$ and $\mathcal{L}(y_1)[c_b] = 1$. With reference to Figure 4b, in each component T_i, $1 \leq i \leq j$, there will be a vertex v_i so that (x, v_i) is an edge in T'. Since T_i is non-polymorphic in c_b, we introduce an edge (x_0, v_i) or (x_1, v_i) in T, depending on whether $\mathcal{L}(v)[c_b] = 0$, or $\mathcal{L}(v)[c_b] = 1$, respectively. Similarly, each component from T_{j+2} to T_k are connected to either y_0 or y_1 by an edge, as shown in Figure 4b. If T_{j+1} is non-empty, there will be vertices v_1 and v_2 in T_{j+1} so that (x, v_1) and (y, v_2) are edges in T'. If $\mathcal{L}(v_1)[c_b] = 0$, we can introduce the edges (x_0, v_1) and (y_0, v_2) in T. If $\mathcal{L}(v_1)[c_b] = 1$, we can introduce the edges (x_1, v_1) and (y_1, v_2) in T. If T_{j+1} is empty (i.e., if x and y are adjacent in T'), we can arbitrarily introduce either the edge (x_0, y_0) or (x_1, y_1) in T. Therefore, all the edges in T' can be inserted back into T in addition to the two edges labeled with c_b. Every row in M can be mapped to a vertex in T, and hence T is an H1-NPP for M. This proves that the existence of the two vertices x and y is a sufficient condition for the matrix M to admit an H1-NPP.

To prove that the existence of the two vertices x and y is a necessary condition, assume that a given matrix M admits an H1-NPP T. We prove that there must be two vertices x and y in T so that $T'_{/\{x,y\}}$ is non-polymorphic

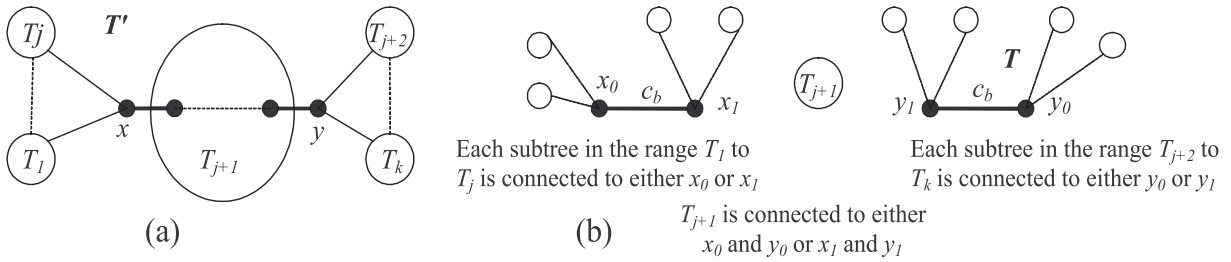


Fig. 4. (a) The perfect phylogeny T', showing $\{T_1, \ldots, T_k\}$, the connected components in $T'/_{\{x,y\}}$; (b) Constructing T from $T'/_{\{x,y\}}$.

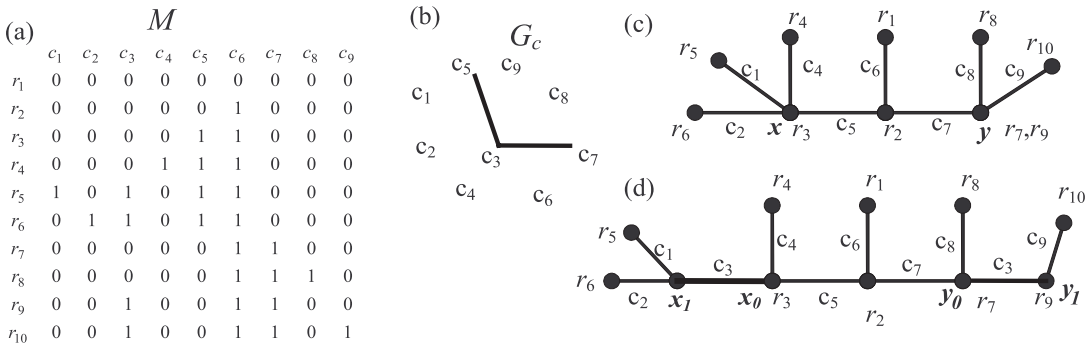


Fig. 5. (a) A matrix M (b) Conflict graph for M (c) Perfect phylogeny T' after removing c_3. (d) The H1-NPP T for M.

in c_b. Since T is an H1-NPP, there must be exactly two edges labeled with c_b in T. Remove the two edges, by collapsing the edges into vertices. Call these vertices x and y. Now obtain the set of trees $T'_{/\{x,y\}}$. Since c_b does not appear as an edge in any of the trees in $T'_{/\{x,y\}}$, c_b is non polymorphic in each component tree. Hence, the existence of the vertices x and y is a necessary condition. ◇

2.1.2 The H1-NPP construction algorithm Theorem 1 and Theorem 2 allow us to determine if a given matrix M admits an H1-NPP and lead to an efficient algorithm to construct a H1-NPP solution for the given matrix M. The heart of the algorithm consists of determining the vertices x and y satisfying Theorem 2 and expanding the nodes into edges labeled with c_b. We have already observed the following properties of the conflict graph G_c:

- The conflict graph G_c for M must have a single non-trivial connected component and there must be at most one vertex with degree greater than one in the conflict graph. If there is any vertex with degree greater than one in G_c, c_b must be that column. If the conflict graph is a single edge connected by two sites, c_b must be one of the two sites.
- Let M' be the matrix produced by removing the column c_b from M. All the sites connected to c_b in the conflict graph must form a path P in the perfect phylogeny T' for the matrix M'.
- Let e_1 and e_2 be the two terminal vertices of the path P in T'. The site c_b should be non-polymorphic in each connected component $T_i \in T'/\{e_1, e_2\}$.

These properties lead to an algorithm for the construction of an H1-IPP for M.

Algorithm Steps

(1) Build the conflict graph G_c for M. If G_c has more than one non-trivial connected component or if there is more than one vertex in G_c with degree greater than 1, M does not admit an H1-NPP. Otherwise proceed to Step 2.

(2) Select the column c_b. c_b will be the column with degree greater than 1 in G_c. If the connected component in G_c is a single edge, arbitrarily pick any of the two vertices that form the edge.

(3) Remove the column c_b from M, and construct a perfect phylogeny T' for the resulting matrix.

(4) Construct the set of columns S_c that are adjacent to c_b in G_c. If M admits an H1-NPP, the columns in S_c must define a path P in T'. Obtain the two terminal ends x and y of this path. If S_c does not define a path in T', M does not admit an H1-NPP.

(5) Check if every connected component in $T'_{/\{x,y\}}$ is non-polymorphic in c_b. If any connected component in $T'_{/\{x,y\}}$ is polymorphic in c_b, M does not admit a perfect phylogeny.

(6) Expand the vertices x and y into the edges (x_0, x_1) and (y_0, y_1), both labeled with the column c_b. Build the phylogeny T as described in the proof of Theorem 2.

Figure 5 illustrates the algorithm. Figure 5a shows a matrix M with nine sites and ten rows. The conflict graph G_c for M is shown in Figure 5b. From the conflict graph, it is clear that removing column c_3 will result in a perfect phylogeny. The perfect phylogeny T' after removing c_3 is shown in Figure 5c. The site c_3 conflicts with sites c_5 and c_7, Hence the path defined by the edges labeled with c_5 and c_7 should be the path between the two mutations in site c_3. Hence the vertices x and y in Figure 5c must be replaced by the edges (x_0, x_1) and (y_0, y_1) in Figure 5d. In Figure 5c, the edges labeled with c_1, c_2, c_4 and c_5 are incident in x. In Figure 5d, the edges c_1 and c_2 are incident on x_1 and c_4 and c_5 are incident on x_0, because of the state of c_3 in r_5, r_6, r_4 and r_2, respectively. The row r_3 now maps to x_0, since $M[r_3, c_3] = 0$. Similarly the edges out of y in T' are distributed between the vertices y_0 and y_1 in T.

Complexity Analysis

Building the conflict graph G_c takes $O(nm^2)$ time. Finding the connected components in G takes $O(m)$ time using depth-first search. Constructing the

(a)

M	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}
r_1	1	0	0	0	0	0	0	0	0	0	1
r_2	0	1	0	0	0	0	0	0	0	1	0
r_3	0	0	1	0	0	0	0	0	0	1	1
r_4	0	0	0	1	1	0	0	0	0	0	0
r_5	0	0	0	1	0	1	1	0	0	1	1
r_6	0	0	0	1	0	1	0	1	0	1	0
r_7	0	0	0	1	0	1	0	0	1	0	1

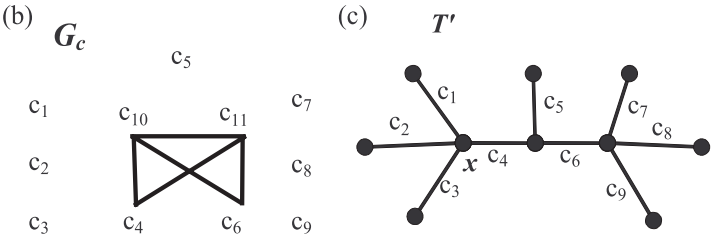


Fig. 6. (a) Matrix M; (b) The conflict graph for the matrix M; (c) The tree T' after removing c_{10} and c_{11}.

perfect phylogeny T' takes $O(nm)$ time, using the opph (Vijaya Satya, 2005, 2006) algorithm. The mapping $\nu(r)$ of each row in M to a vertex in T' takes $O(n)$ space and $O(nm)$ time. Finding the two vertices x and y takes $O(nm)$ time. Building and checking each component in $T'_{/\{x,y\}}$ for being non-polymorphic in c_b takes $O(m)$ time. The overall complexity of the algorithm is thus entirely dominated by the construction of the conflict graph G_c and hence is $O(nm^2)$.

2.1.3 Multiple homoplasy events in a single site An extension of the H1-NPP problem is the case when multiple homoplasy events within the same site are allowed. This situation occurs quiet frequently with true haplotype data. For example, the site 16519 in human mtDNA is expected to have mutated multiple times. We call this problem the H$(1,q)$-NPP problem. Formally, the H$(1,q)$-NPP problem is to construct a phylogeny for the input taxa in which a single site has mutated at most $q+1$ times, where q is an integer greater than 0.

The solution to the H$(1,q)$-NPP problem is an obvious extension of the solution to the H1-NPP problem. As before, the conflict graph G_c for M must have a single connected component, and there should be a single site c_b with degree greater than 1 within this connected component. We can build a perfect phylogeny T' for the matrix M' obtained by removing the column c_b from M. Now, we need to find if there are $q+1$ (or fewer) vertices in T' so that expanding each one of these $q+1$ vertices into an edge labeled with c_b will result in a phylogeny T for M. This can be done by testing all possible combinations of $q+1$ vertices in T' to check if they can lead to an H$(1,q)$-NPP solution. A set $\mathcal{Q}$ of $q+1$ vertices admits an H$(1,q)$-NPP solution if each component in $T'_{/\mathcal{Q}}$ is non-polymorphic in c_b. For any set of vertices $\mathcal{Q}$, this can be tested in $O(m)$ time. We repeat this procedure for values of q starting from 1 to a given maximum value k for q. There are exactly m vertices in T', and there are $\binom{m}{q+1} \cong m^{q+1}$ ways in which $q+1$ vertices can be selected from the m vertices. Therefore, in theory, the complexity of the algorithm is $O(nm^2 + m^{q+2})$ for a given q.

In practice, however, the algorithm can be implemented to run much faster. The following observations reduce the search space significantly:

- If two rows r_1 and r_2 in M with $M[r_1, c_b] = 0$ and $M[r_2, c_b] = 1$ both map to the same vertex z in T', then we call the vertex z as a *polymorphic vertex* with respect to c_b. For obvious reasons, all polymorphic vertices in T' must be expanded into edges labeled with c_b in any H$(1,q)$-NPP for M. Let V_p be the set of polymorphic vertices in T' with respect to c_b.
- Let S_c be the set of sites in G_c that are adjacent to c_b. Each one of the $q+1$ vertices selected for expansion must be incident on an edge labeled with a site in S_c. Therefore, the $q+1$ vertices have to be selected out of l vertices, where $l \leq m$ is the number of distinct vertices in T' that are incident on a edge labeled with a site in S_c. In general, if the degree of c_b in G_c is d, l will be less than or equal to $2d$. Let V_a be the set of vertices in T' that are incident on an edge in S_c.
- Let T_c be the subtree(or forrest) in T' formed exclusively by the sites in S_c. All the leaves of T_c must always be selected for expansion into edges labeled with c_b. Let V_l be the leaves of T_c in T'.

Let $m_c = |V_a|$, and let $m_g = |V_p \cup V_l|$. The actual number of sets $\mathcal{Q}$ that need to be searched is given by $\binom{m_c - m_g}{q+1-m_g}$. Hence, for any matrix M, q will be greater than or equal to $m_g - 1$.

2.1.4. Allowing homoplasy events in multiple sites Extending the problem even further, we define the H(p,q)-NPP problem. An H(p,q)-NPP is a phylogeny in which at most p sites have homoplasy events, with at most q homoplasy events in each site. The conflict graph in this case will have multiple connected components and/or multiple vertices with degree greater than 1.

Let $G_c\prime$ be the graph obtained by removing all degree-0 vertices from G_c. If the matrix M is to admit an H(p,q)-NPP, G_c must have a vertex cover with size less than or equal to p. If such a vertex cover $\mathcal{C}$ is found, removing the vertices in $\mathcal{C}$ from $G_{\mathcal{C}}\prime$ will result in a graph with no non-trivial connected components. We will be able to construct a perfect phylogeny T' for the vertices in $S - \mathcal{C}$. Once T' is constructed, adding any site in $\mathcal{C}$ to T' is an H$(1,q)$-NPP problem.

A necessary (but not sufficient) condition for the existence of an H(p,q) solution is that for each site $i \in \mathcal{C}$, the set of sites $\{S - \mathcal{C}\} \cup \{i\}$ must have a H$(1,q)$ solution. However, adding multiple sites in $\mathcal{C}$ to T' is a more difficult problem. Even if each of the p sites in $\mathcal{C}$ can be added to T' to form H$(1,q)$-NPPs, it does not necessarily imply that the matrix M has an H(p,q)-NPP solution. For example, refer to Figure 6. The conflict graph for matrix M in Figure 6a is shown in Figure 6b. The tree T' after removing c_{10} and c_{11} is shown in Figure 6c. A H$(1,2)$-NPP can be constructed by adding either c_{10} or c_{11} to T', but there is no H$(2,2)$-NPP that includes both c_{10} and c_{11}.

Therefore, to solve the H(p,q)-NPP problem, we need to determine if there is a way to combine the p individual H$(1,q)$-NPP solutions into a H(p,q)-NPP solution. For each site i in $\mathcal{C}$, let Q_i be the set of vertices in T' which have to be expanded into edges labeled with site i in order to add the site i to T' to form an H$(1,q)$-NPP. For each vertex x in T', let $P_x = \{i | x \in Q_i\}$.

DEFINITION. A site $i \in \mathcal{C}$ is fully specified at a vertex $x \in T'$ with respect to an H$(1,q)$ solution consisting of the vertices Q_i if any one of the following conditions are satisfied:

(1) At least one row in M maps to the vertex x.

(2) The vertex x is in a connected component $T_x \in T'_{/Q_i}$, and at least one row in M maps to a vertex in T_x.

Let x and y be two vertices that are adjacent to each other in T'. We define that the two vertices x and y are *pair-wise independent* with respect to a set of H$(1,q)$ solutions for the sites in $\mathcal{C}$ if all of the following conditions are satisfied:

(1) Every site $i \in P_x$ is fully specified with respect to Q_i at the vertex y

(2) Every site $j \in P_y$ is fully specified with respect to Q_j at the vertex x.

(3) $|P_x \cap P_y| = 0$.

A vertex x in T' is defined to be *isolated* (w.r.to the given set of H$(1,q)$ solutions) if x is pair-wise independent with all the vertices adjacent to it.

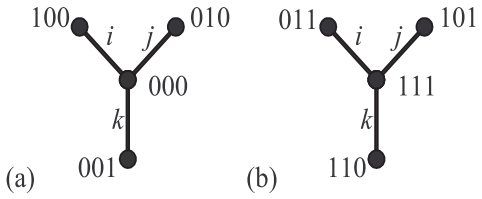


Fig. 7. An example of phylogenies (a) T_x and (b) T_y that must replace two adjacent vertices x and y when x and y are not independent. The node labels of each node over three sites i, j and k are shown.

Each vertex x in $\mathcal{T}'$ must be replaced by a phylogeny $\mathcal{T}_x$ over the sites in P_x. The phylogeny $\mathcal{T}_x$ should be a phylogeny where the taxa include the following:

- The states of the sites in P_x in each row (if any) of M that map to the vertex x.
- For each site y adjacent to x, the state of the sites in P_x at the vertex y.

For example, the vertex x in Figure 6 should be replaced by a phylogeny T_x over the sites $\{c_{10}, c_{11}\}$, where the taxa are $\{00, 01, 10, 11\}$.

When the vertex x is isolated, it can be trivially shown that the following conditions hold true:

(1) All the node labels that must label some node in the phylogeny T_x are known.

(2) For any vertex y adjacent to x, there will be a vertex u in $\mathcal{T}_x$ and a vertex v in $\mathcal{T}_y$ such that $\mathcal{L}(u) = \mathcal{L}(v)$. Therefore, the edge (x, y) in $\mathcal{T}'$ can be replaced by the edge (u, v) in a phylogeny that includes all the vertices in $\mathcal{C}$, and edge (u, v) will not require any more mutations than the edge (x, y).

When any vertex x in $\mathcal{T}'$ is not isolated, and/or if $\mathcal{T}_x$ is not a perfect phylogeny, the H(p, q)-NPP problem is quiet complicated. The phylogenies $\mathcal{T}_x$ and $\mathcal{T}_y$ that replace adjacent vertices will be interdependent, and replacing the edge (x, y) with an edge between some node in $\mathcal{T}_x$ and some node in $\mathcal{T}_y$ might incur additional cost. For example, refer to Figure 7. Let x and y two vertices adjacent to each other with $|\mathcal{P}_x \cap \mathcal{P}_y| = 3$. Let i, j and k be the sites that are common in $\mathcal{P}_x$ and $\mathcal{P}_y$, and let $\mathcal{T}_x$ be the phylogeny shown in Figure 7a and $\mathcal{T}_y$ be the phylogeny shown in Figure 7b. As there are no common vertices in $\mathcal{T}_x$ and $\mathcal{T}_y$, connecting a vertex in $\mathcal{T}_x$ to a vertex in $\mathcal{T}_y$ requires at least one additional mutation in the sites i, j or k.

We leave the unrestricted H(p, q)-NPP problem as an open problem. However, when the following conditions are satisfied, there is a simple solution to the H(p, q)-NPP problem:

- Each vertex in $\mathcal{T}'$ is isolated with respect to the given set of H$(1, q)$ solutions.
- For each vertex x in $\mathcal{T}'$, the phylogeny $\mathcal{T}_x$ that must replace the vertex x is a perfect phylogeny.

When the above two conditions are satisfied, each vertex x can be simply replaced by the perfect phylogeny T_x. As x is isolated, each edge incident on the vertex x in $\mathcal{T}'$ can be replaced by an edge incident on some vertex in $\mathcal{T}_x$, without incurring any additional cost.

Complexity

Finding all vertex covers in G_c with size at most p takes exponential time with respect to p. Assuming the size of G_c is $O(m)$, finding all such vertex covers takes $O(m^{p+1})$ time. For each vertex cover, we need to construct the initial perfect phylogeny T', and find a H$(1, q)$-NPP solution for each site in $\mathcal{C}$. If the set of H$(1, q)$-NPP solutions satisfy the conditions described above, replacing each vertex in T' by a perfect phylogeny takes $O(np)$ time. Hence the over all complexity of the restricted version of the problem is $O(nm^2 + m^{p+1} + \eta p m^{q+2})$ time, where η is the number of distinct vertex covers of G_c with size less than or equal to p.

Special scenarios

A special situation arises when each non-trivial connected component in G_c has at most one site with degree greater than 1. In that case, p will be equal to the number of non-trivial connected components in G_c. The set $\mathcal{C}$ is fixed. This reduces the problem to p completely independent H$(1, q)$-NPP problems that can be solved in $O(nm^2 + pm^{q+2})$ time. In general, each connected component in G_c that is either a single edge or involves a single vertex with degree greater than 1 will reduce the effective value of p by 1.

2.2 Near-Perfect Phylogeny Haplotyping

In case of the NPPH problem, the input is a set of genotypes. The aim in general is to construct a set of haplotypes that are the most likely explanation for the given set of genotypes. Parsimony is widely accepted as the most accurate criterion to reconstruct the phylogeny. Therefore, the aim is to obtain, out of all possible explanations for the given genotypes, the set of haplotypes that admit a phylogeny with the least number of recurrent mutations.

2.2.1 The H1-NPPH problem We formally state the H1-NPPH problem as follows. We are given an $n \times m$ genotype matrix A over the alphabet $\{0, 1, 2\}$. Each row in A represents a genotype. As before, the columns represent SNP sites. The aim is to construct a $2n \times m$ haplotype matrix M such that:

(1) Each row r in A is a result of combining the rows r and r' in M

(2) The matrix M admits an H1-NPP.

The solution to the H1-NPPH problem is very similar to that for the H1-NPP problem, except that it might not be possible to fully construct the conflict graph for a genotype matrix. In a genotype matrix A, an ordered pair of values (a,b), $a \in \{0, 1\}$, $b \in \{0, 1\}$ is in $I(i, j)$ for a pair of columns (i, j) if

(1) There is a row r in A such that $A[r, i] = a$ and $A[r, j] = b$,or

(2) $A[r, i] = a$ and $A[r, j] = 2$, or

(3) $A[r, i] = 2$ and $A[r, j] = b$.

If two columns i and j are '2' in some genotype, the states of i and j in the two haplotypes for the genotype could be either $\{(0, 0), (1, 1)\}$ or $\{(0, 1), (1, 0)\}$. Therefore, we might not be able to completely specify $I(i, j)$. $I(i, j)$ can be completely specified only in two situations: when $|I(i, j)| = 4$ because of rows in A in which either the column i or the column j is not '2', or when there are no rows in A in which both i and j are '2'. Hence, though we might be able to construct some edges in the conflict graph in G_c, we might not be able to construct all the edges in G_c. Therefore, we need other ways to find the column c_b that has a recurrent mutation. One obvious procedure for finding c_b is to remove each column from A, and check if the rest of the matrix admits a perfect phylogeny. If we can find such a column c_b, then there might be a H1-NPPH solution for A. This is the procedure used in Song *et al.* (2005) to find the column c_b. We adopt the same procedure to find c_b. Then, we propose our new algorithm to construct H1-NPPH solution.

Once the column c_b is found, we can build the perfect phylogeny T' for the matrix A' obtained by removing c_b from A. In general, the matrix A' might have multiple perfect phylogenies. Chung and Gusfield (2002) have empirically shown that the likelihood for the phylogeny being unique increases quickly with the number of genotypes. In the following, we assume that A' has a unique perfect phylogeny T'. If A' admits multiple perfect phylogenies, the following procedure has to be repeated for each such perfect phylogeny.

Using the phylogeny T', we construct the haplotype matrix M' for A'. We denote the rows of A' by $r_1, r_2, \ldots, r_n$ and the corresponding pairs of rows in M' as $r_1, r'_1, r_2, r'_2, .., r_n, r'_n$. The matrix M should now be built by adding

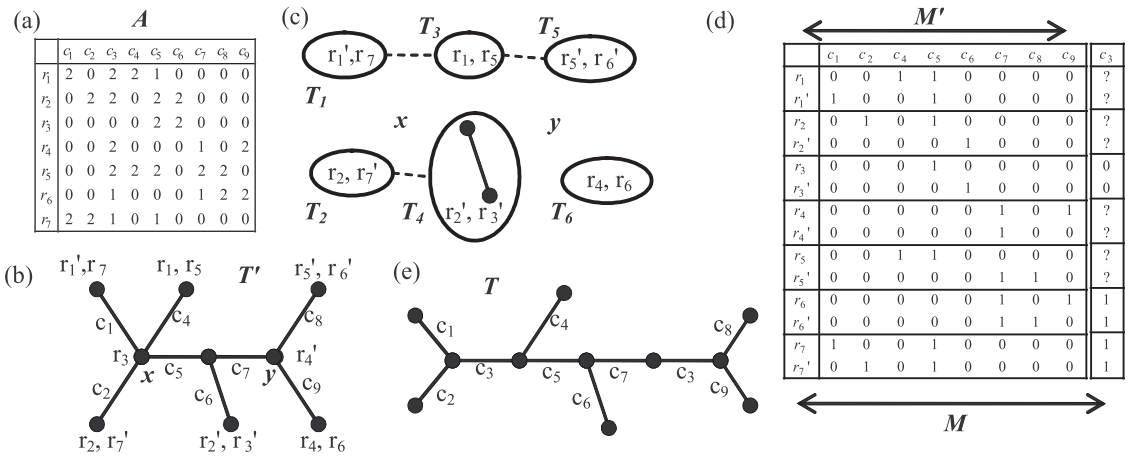


Fig. 8. (a) Matrix A; (b) The tree T′ (c) Components in T′/$_{\{x,y\}}$ overlaid with the edges in G$_a$; (d) Matrices M' and M; (e) The H1-NPP T for the matrix M.

the column c_b to M'. We can also assign values to some rows in column c_b of the matrix M. In a row r_i of A, if $A[r_i, c_b]$ is either 0 or 1, then both the haplotypes for this row will also be either 0 or 1, respectively, in column c_b. We can then set $M[r_i, c_b] = M[r'_i, c_b] = A[r_i, c_b]$. When $A[r_i, c_b] = 2$, we know that $M[r_i, c_b] = \overline{M[r'_i, c_b]}$, but we can not determine which one of them must be 0 for M to admit an H1-NPP. We call such a pair of rows (r_i, r'_i) in M as an *ambiguous pair*. Thus the problem of determining whether A admits an H1-NPP solution reduces to determining whether there is an assignment of values to each such ambiguous pair so that matrix M admits an H1-NPP.

Each row in M'(and hence in M) can be mapped to a vertex in T'. As in the H1-NPP case, we represent this mapping using the notation $\nu(r_i) = v$, where r_i is a row in M, and v is a vertex in T'. For any vertex v in T', zero or more rows in M can map to vertex v.

The underlying idea of our algorithm is based on Theorem 1. We need to identify two vertices x and y, if they exist, such that each connected component in $T'_{/\{x,y\}}$ is non-polymorphic with respect to c_b. We will show how to use this property to actually obtain an assignment of values to each ambiguous pair of rows in M. We arbitrarily choose two vertices x and y in T' and construct a graph $G_a = (V, E)$, where the vertices in V correspond one-to-one to connected components in $T'_{/\{x,y\}}$. For each ambiguous pair (r_i, r'_i) in M, we know that $M[r_i, c_b] = \overline{M[r'_i, c_b]}$. Therefore, if $\nu(r_i)$ is in a component T_i, and $\nu(r'_i)$ is in T_j, we add the edge (v_i, v_j) to E. As each connected component T_i has to be non-polymorphic in c_b, if any unambiguous row r_j maps to a vertex in T_i, we assign the value $M[r_j, c_b]$ to the vertex v_i. Since the value of $M[r_j, c_b]$ is either 0 or 1, we can imagine these two values to represent two 'colors'. Thus, if the chosen pair of vertices $\{x, y\}$ leads to a valid assignment of values to the ambiguous pairs of rows, each connected component in G_a should be two-colorable with the coloring scheme of vertices in G_a as described. Intuitively, a valid two coloring is possible only if the following is true: Let R_0 be the set of rows in M such that $M[r, c_b] = 0$ and similarly let R_1 be the set of rows in M such that $M[r, c_b] = 1$. Then each component G_a has a valid two coloring if and only if for each $T_i \in T'_{/\{x,y\}}$, $R(T_i)$ is a subset of either R_0 or R_1.

If G_a is two colorable given the current coloring of the vertices, each un-colored vertex in G_a can be assigned a color (value) of 0 or 1. When a vertex v_i is assigned a value $a \in \{0, 1\}$, we can assign $M[r, c_b] = a$ for every row r such that $\nu(r)$ is in T_i and $M[r, c_b]$ is un-assigned. After every unknown entry in column c_b of M is filled like this, each connected component $T_i \in T'_{/\{x,y\}}$ will be non-polymorphic in c_b, and hence T' can be converted into an H1-NPP T for M.

Figure 8 shows each step of the procedure. A matrix A is shown in 8a. The perfect phylogeny after removing column c_3 from A is shown in Figure 8b. The matrices M' and M, constructed through T' are shown in Figure 8d. The components in $T'_{/\{x,y\}}$ are shown in Figure 8c. Since the rows r_1 and r'_1 in M form an ambiguous pair, components T_1 and T_3 in G_a are connected. Similarly, components T_2 and T_4 will be connected due to the ambiguous pair (r_2, r'_2), and components T_3 and T_5 are connected due to ambiguous pair (r_5, r'_5). These edges are shown using dashed lines in Figure 8c. Though the rows r_4 and r'_4 also form an ambiguous pair, no edge is added to G_a since one of them (r'_4) maps to the vertex y. Since y will be expanded into two vertices y_0 and y_1, r'_4 can map to any of the two vertices y_0 and y_1, and hence the pair of rows (r_4, r'_4) does not impose any restriction on the coloring of the vertices in G_a. Components T_1, T_2, T_4, T_5 and T_6 can similarly be assigned a color of 1 because of the rows r_7, r'_7, r'_3, r'_6 and r_6, respectively. The connected component T_3 can not directly be assigned any color, since no unambiguous row maps to it. It can be seen that G_a is two-colorable, and the only possible coloring is to assign color 0 to T_3. The final H1-NPP T is shown in Figure 8e.

The fundamental problem now is how to find the two sites x and y in T'. In case of the H1-NPP problem in Section 2.1, the conflict graph G_c could be constructed, RMP could be deduced from G_c, and the two vertices x and y could be directly selected as the terminal ends of RMP. In case of the H1-NPPH problem, since we can not construct the conflict graph completely (unless in very obvious special scenarios), we must exhaustively search for the vertices by checking each pair of vertices in T'. Since there are exactly m vertices in T', there will $O(m^2)$ pairs of vertices that we need to check.

For each pair of vertices, the graph G_a can be constructed in $O(n + m)$ time, allowing parallel edges. Since there are at most $O(n)$ edges in G_a (at most one for each row in A), the connected components in G_a can be identified in $O(n + m)$ time using depth-first search. Two-coloring of G_a can be obtained in $O(n + m)$ time using breadth-first search. Hence, the overall complexity of the algorithm is $O(m^2(n + m))$.

It might seem that the $O(n^4)$ algorithm of [Song *et al*. (2005) might perform better if $m > n$. However, m can never be greater than $O(n)$ without having duplicate columns in M. This is because even if each of the $2n$ haplotypes are distinct, there can be no more than $4n - 4$ edges in the tree. With only one homoplasy event, each column except c_b has to label a distinct edge, and hence there can be at most $4n - 3$ distinct columns in the matrix M. If the matrix M has more than $4n - 3$ distinct columns, it will not admit an H1-NPP.

On the other hand, n can be as high as $O(m^2)$. Hence, our algorithm has better time-complexity than the previous $O(n^4)$ algorithm for any value of n and m.

2.2.2 Making use of the conflict graph The conflict graph provides useful information that can be utilized to speed up the above algorithm. Even though it might not be possible to build the conflict graph completely, we can

Table 1. Comparison of our H1-NPPH method with PHASE for different datasets. The running times are on Pentium 3.2 GHz PC

Test case	Our H-1 NPPH algorithm			PHASE		
	Std. error	% of mis-phased 2's	Run-time	std. error	% of mis-phased 2's	Run-time
50×50	0.0116	0.157%	0.01s	0.0138	0.269%	109s
100×50	0.0054	0.064%	0.016s	0.0046	0.065%	268s
50×100	0.011	0.105%	0.031s	0.0156	0.214%	497s
100×100	0.0048	0.046%	0.047s	0.011	0.136%	874s

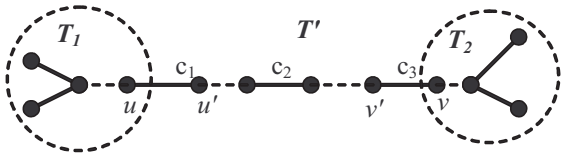


Fig. 9. Any solution must involve a vertex from T_1 and a vertex from T_2.

make use of what is available of the conflict graph in order to reduce the $O(m^2)$ search space of the pairs of vertices.

Inferring the recurrent mutation path

From the discussion in Section 2.1, it is clear that the set S_c of sites adjacent to c_b in the conflict graph must all lie in a path in T'. Let $S_c = \{c_1, c_2, c_3\}$, and let all three of them lie in a path in T', as shown in Figure 9. If the matrix A admits an H1-NPP, the path between the two vertices x and y that are selected to be expanded must clearly include all the sites in S_c. Therefore, one of them (say, x) has to be in T_1 and the other (say, y) has to be in T_2, as shown in Figure 9. Therefore, the conflict graph can be effectively used to reduce the pairs (x, y) that need to be checked. The following is another interesting result:

LEMMA 1. *The sites in S_c must form a contiguous path in T' if the matrix A admits an H1-NPP.*

i.e, the sites c_1, c_2 and c_3 must form a contiguous path, instead of a broken path as depicted in Figure 9. We do not provide a formal proof for Lemma 1 as we do not directly use it in this paper.

Using the ambiguous pairs more effectively

For any ambiguous pair of rows (r, r') in M, the path between the vertices $\nu(r)$ and $\nu(r')$ must include an edge (in general, an odd number of edges) labeled with c_b. This means that any pair of vertices x and y in T' that are a possible solution must be such that $\nu(r)$ and $\nu(r')$ are not in the same connected component $T_i \in T'_{/\{x,y\}}$. The following lemma states this property formally:

LEMMA 2. *For any two vertices x and y in T' that can be expanded to form a H1-NPPH solution for matrix A, the path between the vertices $\nu(r)$ and $\nu(r')$ for every ambiguous pair (r, r') must include the vertex x or y or both.*

PROOF. Let there be an ambiguous pair (r, r') in M so that the path in T' between the two vertices $\nu(r)$ and $\nu(r')$ does not include both x and y. This means that the vertices $\nu(r)$ and $\nu(r')$ are in the same connected component $T_i \in T'_{/\{x,y\}}$. Since $M[r, c_b] = \overline{M[r', c_b]}$, this implies that T_i is polymorphic with respect to c_b. Hence, there must be an edge within T_i labeled with c_b in addition to the two edges labeled with c_b inserted at the vertices x and y. Hence the two vertices x and y can not lead to an H1-NPPH solution for the matrix A. Therefore, for any pair of vertices x and y in T' that can be expanded into an H1-NPPH solution for matrix A, the path between the vertices $\nu(r)$ and $\nu(r')$ for every ambiguous pair (r, r') must include the vertex x or y or both. ◇

Table 2. Properties of the data sets generated

test case	#of datasets (out of 100) that admit a perfect phylogeny	#of datasets admitting H-1 NPPH solutions (with a unique PPH solution for A')
50×50	16	84 (49)
100×50	10	90 (54)
50×100	3	97 (55)
100×100	8	92 (42)

Lemma 2 can be used to avoid checking some vertex pairs. Let R be the set of rows in A such that $A[r, c_b] = 2$ for every $r \in R$. Let $R_x \subseteq R$ be the set of rows in A such that, for every $r \in R_x$, the path between the vertices $\nu(r)$ and $\nu(r')$ in T' includes the vertex x in T'. Similarly, let R_y be the corresponding set of rows for the vertex y in T'. The pair of vertices x and y can not be a solution unless $R = R_x \cup R_y$.

2.2.3 The H(1, q)-NPPH problem The solution for the H$(1,q)$-NPPH problem is a simple extension to the solution for the H1-NPPH problem. All the discussion above applies to H$(1,q)$-NPPH problem, with the only difference being that instead of finding a pair of vertices x and y, we need to find a set of $q+1$ vertices $\mathcal{Q}$ so that $\mathcal{T}'$ can be converted into an H$(1,q)$-NPP T by expanding each one of $q+1$ vertices in $\mathcal{Q}$ into an edge labeled with c_b.

In case of the H$(1,q)$-NPP problem, we could use G_c to narrow down the possible sets of vertices for $\mathcal{Q}$. We can not do the same thing here, since G_c is not complete. Therefore, we need to try all-possible sets of vertices of size $q+1$. There are $\binom{m}{q+1}$ such possible sets of vertices. For each set, testing if the set of vertices form a solution is identical to the procedure for the H1-NPPH problem—we build the graph G_a in which each vertex represents a connected component in $T'_{/\mathcal{Q}}$. As before, two vertices v_i and v_j have an edge between them if there is an ambiguous pair (r, r') in M so that the vertex $\nu(r)$ is in v_i and the vertex $\nu(r')$ is in v_j. We need to test if the graph G_a is two-colorable. As in the case of the H$(1,q)$-NPP problem, This algorithm can be implemented to run in $O(nm^2 + m^{q+1}(n+m))$ time.

2.2.4 The H(p, q)-NPPH problem Like the H(p,q)-NPP problem, the H(p,q)-NPPH problem can be viewed as a set of H$(1,q)$-NPPH problems. We first need to find a set of p columns $\mathcal{C}$ so that the matrix A' obtained by removing the columns in $\mathcal{C}$ from A has a perfect phylogeny T'. Once T' is constructed, we can solve for each of the sites in $\mathcal{C}$ as an H$(1,q)$-NPPH problem. The haplotype matrix M can be constructed for a given set of H$(1,q)$-NPP solutions, and the H(p,q)-NPPH problem on the matrix A will be equivalent to the H(p,q)-NPP problem on the matrix M. However, if any site $i \in \mathcal{C}$ has multiple H$(1,q)$-NPP solutions, there will be multiple such matrices M, and the matrix A will admit an H(p,q)-NPP if

any one of those matrices admit a H$(1,q)$NPP. The time complexity of the algorithm will be similar to that of the H(p,q)-NPP algorithm.

3. RESULTS

We have implemented our algorithm for the H1-NPPH problem in C++. In this section, we compare the performance of our algorithm to that of PHASE (Stephens *et al*., 2001) using simulated data. To generate the simulated data, we follow the same procedure as in Song *et al*. (2005). We first generate homoplasy-free haplotype matrices with minimum allele frequency (MAF) $\geq 2\%$ using the program MS (Hudson, 2002). In each matrix, we introduce a homoplasy column by randomly selecting two vertices in the perfect phylogeny for the dataset and expanding the two vertices into edges labeled with the newly introduced column. We ensure that the newly introduced column has a MAF $\geq 2\%$ by selecting two non-adjacent vertices for expansion. Finally, we construct the genotype matrix by pairing consecutive rows in the haplotype matrix.

The results are summarized in Tables 1 and 2. We provide two measures of accuracy. The first measure, the standard error, is the ratio of the genotypes that are incorrectly inferred to the total number of genotypes in the data set. The second measure is simply the percentage of mis-phased 2s. We used 100 datasets for each problem size. The run-times and error-rates shown are averages for the hundred datasets.

4 DISCUSSION

The algorithms and problem formulations we introduced here are applicable in a wide a variety of problems encountered in genome variation studies and population genetics. With the help of simulated data, we demonstrated that the algorithms are applicable and practical in case of the haplotype inference problem. We believe that these algorithms will also be practical for phylogenetic reconstruction problems in general. Specifically, the algorithms will be extremely useful for inferring phylogenies for haploid genomes, like mtDNA and the human Y-chromosome.

REFERENCES

Bafna,V., Gusfield,D., Lancia,G. and Yooseph,S. (2002). Haplotyping as perfect phylogeny: a direct approach. Technical Report CSE-2002-21 Department of Computer Science, The University of California at Davis.

Bonizzoni,P., Vedova,G.D., Dondi,R. and Li,J. (2003) The haplotyping problem: and overview of computational models and solutions. *Journal of Computer Science and Technology*, **18**(6), 675–688.

Chung,R.H. and Gusfield,D. (2002). PPH—a program for deducing haplotypes that fit a perfect phylogeny. Technical Report CSE-2002-27 Department of Computer Science, The University of California at Davis.

Clark,A.G. (1990) Inference of haplotypes from pcr-amplified samples of diploid populations. *Mol. Biol. Evol*., **7**, 111–122.

Ding,Z., Filkov,V. and Gusfield,D. (2005) A linear time algorithm for the perfect phylogeny haplotyping (pph) problem. In *Proceedings of RECOMB*, MIT, Cambridge, MA.

Fernandez-Baca,D. and Lagergren,J. (2003) A polynomial time algorithm for near-perfect-phylogeny. *SIAM Journal of Computing*, **32**(5), 1115–1127.

Gusfield,D. (2002) Haplotyping as perfect phylogeny: conceptual framework and efficient solutions. In *Proceedings of RECOMB*.

HapMapConsortium (2003) The international hapmap project. *Nature*, **426**, 789–796.

Halperin,E. and Karp,R. (2004) Perfect phylogeny and haplotype assignment. In *Proceedings of RECOMB*.

Hudson,R. (2002) Generating samples under the wright-fisher neutral model of genetic variation. *Bioinformatics*, **18**, 337–338.

Hudson,R. and Kaplan,N. (1985) Statistical propertied of the number of recombination events in the history of a sample of dna sequences. *Genetics*, **111**, 147–165.

Kimmel,G. and Shamir,R. (2005) The incomplete perfect phylogeny problem. *J Bioinform Comput Biol*., **3**(2), 1–25.

Liu,Y. and Zhang,C.Q. (2004) A linear solution for haplotype perfect phylogeny problem. In *International Conference on Bioinformatics and its Applications (ICBA)*, Nova Southeastern University, Fort Lauderdale, USA.

Song,Y.S., Wu,Y. and Gusfield,D. (2005) Algorithms for imperfect phylogeny haplotyping (ipph) with a single homoplasy or recombination event. In *Proceedings of WABI 2005* pp. 152–164.

Sridhar,S., Dhamdhere,K., Blelloch,G.E., Halperin,E., Ravi,R. and Schwartz,R. (2005). FPT algorithms for binary near-perfect phylogenetic trees. Technical Report CMU-CS-05-181 Computer Science Department, Carnegie Mellon University School.

Stephens,M., Smith,N. and Donnelly,P. (2001) A new statistical method for haplotype reconstruction from population data. *Am J Hum Genet*., **68**, 978–989.

Vijaya Satya,R. and Mukherjee,A. (2005) An efficient algorithm for perfect phylogeny haplotyping. In *Proceedings of CSB2005* pp. 103–110, Stanford, CA.

Vijaya Satya,R. and Mukherjee,A. (2006) An optimal algorithm for perfect phylogeny haplotyping. *Journal of Computational Biology*, **13**(4), 897–928.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e523–e529
doi:10.1093/bioinformatics/btl241

SNP Function Portal: a web database for exploring the function implication of SNP alleles

Pinglang Wang[1], Manhong Dai[1], Weijian Xuan[1], Richard C. McEachin[2], Anne U. Jackson[3], Laura J. Scott[3], Brian Athey[2], Stanley J. Watson[1] and Fan Meng[1,2,*]

[1]Molecular and Behavioral Neuroscience Institute and Department of Psychiatry, University of Michigan, Ann Arbor, MI 48109, [2]National Center for Integrative Biomedical Informatics, University of Michigan, Ann Arbor, MI 48109 and [3]Biostatistics Department, School of Public Health, University of Michigan, Ann Arbor, MI 48109

ABSTRACT

Motivation: Finding the potential functional significance of SNPs is a major bottleneck in understanding genome-wide SNP scanning results, as the related functional data are distributed across many different databases. The SNP Function Portal is designed to be a clearing house for all public domain SNP functional annotation data, as well as in-house functional annotations derived from different data sources. It currently contains SNP functional annotations in six major categories including genomic elements, transcription regulation, protein function, pathway, disease and population genetics. Besides extensive SNP functional annotations, the SNP Function Portal includes a powerful search engine that accepts different types of genetic markers as input and identifies all genetically related SNPs based on the HapMap Phase II data as well as the relationship of different markers to known genes. As a result, our system allows users to identify the potential biological impact of genetic markers and complex relationships among genetic markers and genes, and it greatly facilitates knowledge discovery in genome-wide SNP scanning experiments.

Availability: http://brainarray.mbni.med.umich.edu/Brainarray/Database/SearchSNP/snpfunc.aspx

Contact: mengf@umich.edu

1 INTRODUCTION

A Single Nucleotide Polymorphism (SNP) is a DNA sequence variation at a single nucleotide level. It is estimated that SNPs occur once per 100∼300 bases in the human genome. The dramatic increase in genotyping efficiency in the last couple of years has made large-scale high density genome-wide SNP association analysis practical for many research groups. It can be expected that these genome-wide SNP association studies will identify many SNP alleles related to various complex disorders. Identifying the causative relationships between many disease predisposing alleles and the corresponding disorders will be a major challenge.

Understanding the potential biological implications of SNP alleles will be more difficult than understanding influences of dynamics at gene expression or protein levels. This is because proteins are direct players in various molecular, cellular or higher level pathophysiological processes. In addition, though mRNAs *per se* do not directly participate in these processes under most circumstances, the regulation of mRNA expression has proven to be a critical mechanism in biological function regulation. Consequently, as a working hypothesis, mRNA and protein variants can often be treated as indicators of functional changes in the corresponding genes.

Genotype data, on the other hand, have much more complex and indirect relationships with genes and proteins. In the simplest scenario, a SNP allele may cause a key amino acid residue change in a critical protein functional domain and then alter protein function. Unfortunately, in many situations we do not fully understand the function or impact of a SNP allele. Even if a SNP allele causes an amino acid residue alteration in a protein, this substitution does not necessarily lead to biologically significant consequences. In fact, most SNPs are not even in the coding sequences of genes. They may influence biological processes in many conceivable ways: reduce transcription factor binding affinity to the promoter region, alter a microRNA binding site, change mRNA stability, modify the RNA splicing pattern, destroy an internal ribosomal binding site, etc. Given the complexity of the way that a SNP allele may influence the function of a protein, it is highly desirable to have a comprehensive database where researchers can easily access the most up-to-date SNP functional annotations.

Although there are several efforts on the functional annotation of SNPs, the coverage of existing commercial or public domain efforts is far from complete. So far these groups are each focusing on a narrow set of annotations, such as protein domain change, splicing, location of SNP in relationship to known genes, etc. (Bao & Cui, 2005; Kang *et al*., 2005; Kasprzyk *et al*., 2004; Maglott *et al*., 2005; Reumers *et al*., 2005). Also it is often hard for the existing software to accommodate new or user created SNP functional annotations. As a result, the task of finding the potential functional consequences for a SNP allele associated with a given disease requires the exploration of many different data sources. Given the fact that a typical researcher is not likely to be fully knowledgeable about all major SNP annotations created by different research groups, it can be expected that most research groups may not thoroughly explore the functional consequences of the SNPs identified from genome-wide association studies. This is evident from existing publications, as most papers focus on SNPs that affect the coding, promoter, or splicing regions. Very few authors discuss other potential functional consequences such as mRNA stability

*To whom correspondence should be addressed.

and internal ribosomal binding site affinity that may be affected by variants. Such a limited focus, caused by the lack of appropriate tools or resources for understanding the full implications of SNP alleles, significantly hinders generation of testable hypothesis based on genome-wide association analysis results.

In addition, neighboring SNPs usually show different degree of linkage disequilibrium (LD). Consequently, although a SNP allele itself does not cause any functional difference, a tightly linked nearby allele may be the causative allele. With existing tools, a researcher has to go through the HapMap data set to find the LD region for a given population and then go to the dbSNP to find all SNPs in the same region before he/she is able to identify the potential functional impact of a SNP derived from a genome-wide association study. Such a process is very time consuming, so a highly automated procedure is of great assistance.

Furthermore, besides functional annotations at the molecular level (i.e., gene/mRNA/protein), it would be very helpful if an annotation system could associate SNPs with genes related to higher level cellular and pathophysiological processes, since most complex disorders have heterogeneous molecular etiology involving the combined effect of multiple molecular entities. Merely focusing on individual genes does not provide a big picture of all SNPs that may participate in the same cellular or pathophysiological process. In order to understand how heterogeneous molecular factors may lead to similar pathological phenotypes, it is necessary to have higher level functional annotation. This annotation should include pathway, gene ontology and disease association based on various criteria, for groups of SNPs that may have diverse molecular function implications but may collectively influence a common biological process.

Lastly, researchers need to have multiple ways to access the annotation data for different purposes. Of course, the majority of users would like to perform SNP function searches through a friendly web interface and explore various types of functional annotations based on the LD data derived from the HapMap project. However, some researchers may need to download specific annotations not available at other databases (e.g., SNPs overlapping with internal ribosomal entry sites or SNP groups based on KEGG pathways) for integration with their local databases or local algorithms that can take advantage of SNP function information. Even if the same annotation is available from other resources, it will be convenient to have a single source for curated links to various data download sites. Some advanced users may want to incorporate such SNP functional annotation capabilities in the web functions they are building. Consequently, it will also be helpful to establish web services based on WSDL/SOAP standards to enable programmatic access, providing the possibility of building complex web applications using functional annotation data developed by different groups.

The main goal of this work is to build a comprehensive SNP function portal to facilitate the understanding of functional implications of SNP alleles identified in genome-wide association studies. We integrate annotation from different databases and generate functional annotations based on various existing sequence and structure analysis algorithms. We also provide annotation of SNPs related to high level biological functions. We integrate a powerful SNP search function that utilizes the LD data from the HapMap project in the SNP function search process. The portal accepts generic markers including SNPs, genes, microsatellite markers and cytogenetic bands as input. To meet the requirements of different users, we currently provide a web service for identifying all genetically related SNPs, as well as batch annotation data download in multiple formats (text, Excel spreadsheet, etc.). The SNP Function Portal greatly increases researcher's efficiency at SNP function exploration and it will be continuously improved by adding more features and functional annotations.

2 SYSTEM AND METHODS

The SNP Function Portal currently has three main modules: 1) a powerful SNP search function that maps input genetic markers to all physically or genetically related SNPs satisfying user's criteria, based on the HapMap II and dbSNP data 2) a SNP function data integration pipeline that obtains updated annotation data from external sources and generates annotations using existing sequence and structure analysis programs, and 3) a web interface that receives user input, generates a summary report and provides flexible browsing, filtering, sorting and downloading capabilities. The relational SNP database is built on Oracle 10g, with data downloaded and parsed from various sources as well as generated by our internal algorithms. The web function was implemented with the mix of .Net, ASP, JSP and Perl, and is hosted on a Windows 2003 Sever.

2.1 SNP search function

We believe a comprehensive SNP search capability is a necessity for any SNP functional annotation database. Based on our own experience, researchers may start a SNP function query for any type of genetic marker: SNP IDs, STS/microsatellite marker IDs, cytoband/genomic region or even gene/protein identifiers.

For example, researchers frequently want to identify potential functionally important SNPs using differentially expressed gene lists derived from gene expression analysis or disease-related candidate genes identified in literature. As a result, it will be highly desirable to use gene ID as input and to provide users the ability to define SNPs related to a gene based on criteria such as their distance from the gene, whether they share the same LD region with a gene, or locate in the gene promoter.

Furthermore, there is a large body of literature describing the linkage or association of STS/microsatellite markers and cytobands to various diseases. To the best of our knowledge, none of the existing web functions provides direct mapping of these genetic markers to SNPs based on their genomic locations and existing LD data, and such association may be extremely time-consuming to perform manually.

Thus, we developed a convenient SNP search function that will generate a complete list of SNPs based on genetic marker input and SNP filtering criteria provided by users. This is achieved through the integration of the data from the dbSNP, UniSTS, NCBI ideogram, Entrez Gene, NCBI human genome assembly and HapMap II project. In order to enable haplotype-based SNP filtering, we pre-calculated haplotype blocks in the four HapMap sample populations using two different methods. We will update haplotype calculations upon each new HapMap release or NCBI genome assembly release. The rest of the data used in the SNP search function are updated monthly to maintain their concurrence with the related data sources.

2.2.1 SNP functional annotation categories As mentioned previously, although there are a number of public and commercial efforts to provide functional annotations for SNPs, none of them has the desired coverage, and it is not easy to add new annotation to their solutions. In order to provide a comprehensive overview of SNP functional annotations to meet the requirements from different researchers, we collect SNP functional annotations from various sources and organize them into six major categories as shown in Table 1. These categories form a core framework to encapsulate existing and new annotations in our database. The overview of general data sources, organization and flows is described in Figure 1.

Table 1. Functional Annotations for the SNP Function Portal

Genome Level	Conserved Genomic Region
	Repetitive Sequence
Transcript Level	CpG Island
	DNase 1 Hypersensitive Site
	Histone Acetylation Site
	Known Transcription Factor Binding Site
	Predicted Transcription Factor Binding Site
	RNA Polymerase Binding Region
	Splicing donar/acceptor sites
	Intronic and Exonic Splice Enhancer/Silencer
	Branch Site Recognition Sequence
	MicroRNA Transcript
	MicroRNA Binding Site
	RNA Stability
Protein Level	Internal Ribosomal Entry Site
	Known Or Potential Modification Site, Including Phosphorylation, Sulfation, Methylation, Acetylation, Palmitylation, Myristoylation, Glycosylation, etc
	Key Residues Influence
	Protein Domain Structure
	Protein Location Motif: Signal Peptide, Nuclear Localization Signal
	Conserved Protein Domain
	Activity/Binding Center Of Protein
	Protein-Protein Interaction Interface
Pathway/Ontology Level	Gene Ontology
	KEGG Pathway
	BioCarta Pathway
	Molecular Interaction Network
Disease Level	OMIM annotation
	Literature description
	Association Statistics
	Linkage Statistics
	SNP-SNP Interaction Statistics
	Expression Profile
Population Genetics Level	Linkage Disequilibrium Scores
	Haplotype
	Ancestral Allele
	Tajima's D
	Fst Value
	Heterozygosity
	Allele Frequency
	Map Weight

In the left column, six major classes of annotation are shown. Specific annotations are itemized in the right.

2.2.2 Data processing and integration pipeline We have currently processed the majority of SNP functional annotation data listed in Table 1. Besides parsing and integrating SNP annotation data from various sources, we add custom annotation using various sequence analysis methods. The following descriptions focus on procedures used for these custom annotations as well as some special annotation types.

Transcription factor binding site. The vast majority of SNPs occur in noncoding sequences of the genome. Their influences on phenotype may be through biological mechanisms such as transcription factor binding, alternative splicing, etc. To help researchers dissect the impacts of SNP alleles on transcriptional control of gene expression, we analyzed the nucleotide sequences containing SNP alleles and tried to identify the transcriptional factor binding sites they may affect. For each SNP, a pair of sequences containing either the major or minor allele (together with their shared 5' 24 nucleotides and 3' 24 nucleotides) is derived from the dbSNP. Then, each pair of these major allele/minor alleles containing sequences related to a given SNP is analyzed for potential transcription factor binding site matches through sequence alignment, as shown in Figure 2. We create an entry in our annotation database if the two alleles from the same SNP have different match scores with a potential transcription factor binding site.

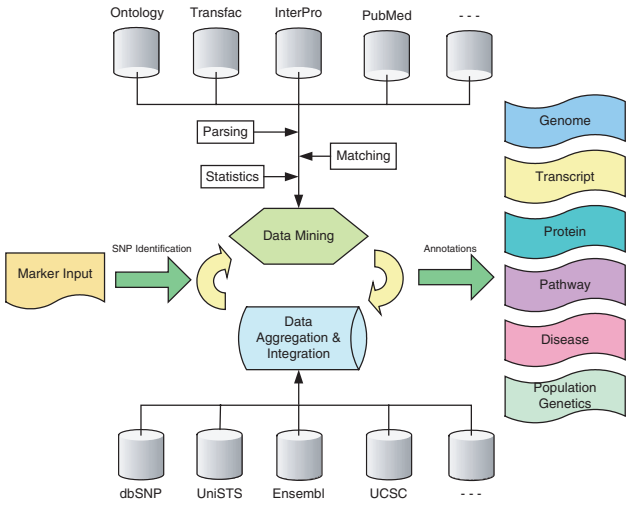


Fig. 1. SNP portal data sources, organization and flows.

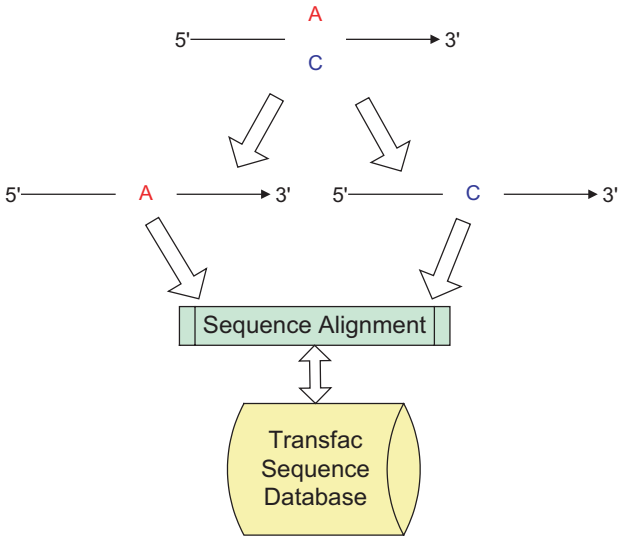


Fig. 2. Analysis of SNP alleles against transcription factor binding sites.

In the current version, we utilize two programs provided with the TRANSFAC Pro database, Match™ (Kel *et al*., 2003) and Patch™, to identify potential transcription factor binding sites in SNP sequences. Match™ uses a library of mononucleotide weight matrices in TRANSFAC to find transcription factor binding sites in our SNP sequence constructs, while Patch™ does so by pattern searching of transcription factor binding sites and the consensus sequences of weight matrices in the TRANSFAC database. For Match™, we choose the parameters to minimize the sum of both false positive and false negative rates. With Patch™, we search against all transcription factor binding sites in the TRANSFAC site table and all

consensus sequences from the TRANSFAC matrix table, and the output cut-off score is set at 70 for balanced accuracy and sensitivity. The combination of these two methods allows researchers to examine all transcription factor binding sites in the TRANSFAC database. However, since the professional version of the TRANSFAC database requires a license, only the transcription factor binding site IDs and their match scores are initially visible to public users.

Protein domain structure. Nonsynonymous SNPs, whose alleles encode different amino acid residues, are most interesting to researchers since they may offer a simple explanation of the biological consequence of different SNP alleles at the protein function level. However, many nonsynonymous SNPs may not cause protein function changes because the corresponding residues may not be in functionally important regions. To help researchers to quickly identify potential causal relationships between nonsynonymous SNP alleles and protein structure changes, we analyze the protein sequence variations caused by the nonsynonymous SNPs against currently known protein domains. Similar to the transcription factor binding site analysis, we first derive a pair of complete protein sequences for each nonsynonymous SNP: one for the major and the other for the minor SNP allele. In the current version, we simply treat all the SNP-caused amino acid residue changes individually, without considering the underlying genetic linkage structure. As a result, we generate multiple protein sequence pairs for proteins having several nonsynonymous SNPs, with each sequence pair containing only one amino acid residue alteration. Next, using the InterProScan (integrated protein domain scanning tool from EBI), we scan those sequence constructs against all the protein domains in InterPro databases, including protein domain/motif data such as ProDom and PRINTS, to identify protein domain matches and the domain match scores. If there is a score difference between two alleles of the same SNP for any matched protein domain, the related SNP will be highlighted in our output and all the corresponding scores will be displayed for researchers' further review. Out of 51,807 nonsynonymous SNPs we identified in 17,488 reference protein sequences, 1,083 SNPs caused protein domain changes based on InterProScan results.

Biological functional categories: Rather than focusing only on the SNP functional consequence at the single molecule level, we also map SNPs to genes, Gene Ontology, Cytoband, KEGG pathway, BioCarta and GenMAPP in order to facilitate the understanding of the impact of SNPs in higher level biological functions. We first map the SNPs to Entrez genes according to their genomic locations, and then the Entrez genes are mapped to different functional categories. Consequently, SNPs are associated with biological function categories in popular gene function annotation databases. In order to help users to identify significantly over-represented function category matches, our system performs an on-the-fly Fisher's Exact Test for each matched function category based on the user provided input SNP list.

Disease association: Free text literature databases such as Medline and OMIM contain extensive information on the relationship between genetic markers and diseases, although most existing genetic study literature focuses on cytoband, microsatellite markers and genes. Since these genetic markers can be easily mapped to nearby SNPs in our system, the large body of free text literature on genetic analysis of diseases provides a rich resource of information for understanding the potential functional significance of SNPs. To effectively utilize the related information in free text literature, we build a free text literature processing pipeline for extracting information on genetic markers and their relationships to diseases, using natural language processing techniques. More than one thousand diseases in OMIM are linked to genetic studies described in the Medline. The corresponding genetic markers and literature links are stored in our database. The related details will be described in a separate manuscript, but interested users may want to try the MarkerInfoFinder (http://brainarray.mbni.med.umich.edu/Brainarray/DataMining/MarkerInfoFinder/default.asp), which is the main product of our genetic marker literature mining project. It enables the search of the Medline database using various genetic marker names directly.

Links to external sources. We integrate downloadable raw data directly related to, or useful for, SNP functional annotation from major public domain databases. However, there are also useful web databases that currently support web-based inquiries only, with no raw data downloadable. For the user's convenience, we provide direct web links to data in these external databases through reference SNP IDs. Researchers can easily navigate to these external web databases for SNP annotations not in our database for additional details. For example, for each SNP, we include direct web links to the dbSNP and the LS-SNP (predictions of protein functional changes due to SNP) from University of California San Francisco (Karchin *et al*., 2005). We continue to add data to support our integrated annotation database. If there are public or custom SNP annotation data that researchers would like to include in our web database, we would either directly integrate the data into our database or provide corresponding web links based on reference SNP IDs.

HapMap data calculation: The identification of blocks of SNPs in the same LD region is critical for the understanding of the functional significance of candidate alleles. We integrate the HapMap Phase II data to support function exploration of all genetically related SNPs based on different LD scores such as r^2, D' and *LOD* score. In addition, we calculate the haplotype blocks with two methods, Confidence Intervals (Gabriel *et al*., 2002), and Four Gamete Rule (Wang *et al*., 2002), using the HaploView package (Barrett *et al*., 2005). The calculation of haplotype blocks is performed on all four populations currently in the HapMap project: European, Chinese, Japanese and African. As a result, researchers can easily find all SNPs genetically related to their input list (e.g., SNP, STS, gene) in these four populations.

2.3 Web functions

Web Interface: Our portal supports searches of functionally related SNPs at one stop. With our search engine, researchers can freely scan the related genomic and genetic regions of their targeted SNPs. Even if some researchers do not have targeted SNP lists, they can search for SNP IDs in genomic regions, genes, functional categories, and pathways of interests through our Search SNP web function (not shown). In addition, our SNP portal supports UniSTS IDs as input. We will automatically match the genetic markers to the closest SNPs and conduct the additional genomic and genetic scanning based on user-defined criteria. Figure 3 shows the search interface for the SNP Function Portal.

In the initial search interface, users need to provide a list of either SNP or UniSTS IDs. They can also find SNPs that they might be interested in through our Search SNP function, which will fill SNPs list input box automatically based on users' criteria. Researchers can indicate the size of neighboring genomic regions for the SNP list they desire. Our search engine will automatically search and identify all the SNPs located in those genomic regions. One may also select gene neighbor to include SNPs in the context of the genes and their promoter regions for the SNPs in their input list. Our search engine will first search for all the Entrez genes that the input SNPs belong to, and then include all the SNPs in the genes as well as their 5' upstream regions users select.

Furthermore, our search function will search and include all the SNPs within user-defined linkage disequilibrium regions through different linkage disequilibrium scores or haplotype blocks. Taking all the SNPs in the related genomic/gene neighbors as a pool, our search engine will identify all SNPs satisfying the user-defined linkage disequilibrium criteria. Finally, the SNPs identified through all the above-mentioned processes will be returned to users in the result page, along with their annotations currently available in our database. The searching process is demonstrated in Figure 4. Samples of annotations are demonstrated in Figure 5 and Figure 6.

2.4 Additional annotation data access methods

While we expect most users to use the SNP Function Portal web interface to explore the functional significance of their chosen genetic markers, we also provide different ways to download data to meet large-scale custom

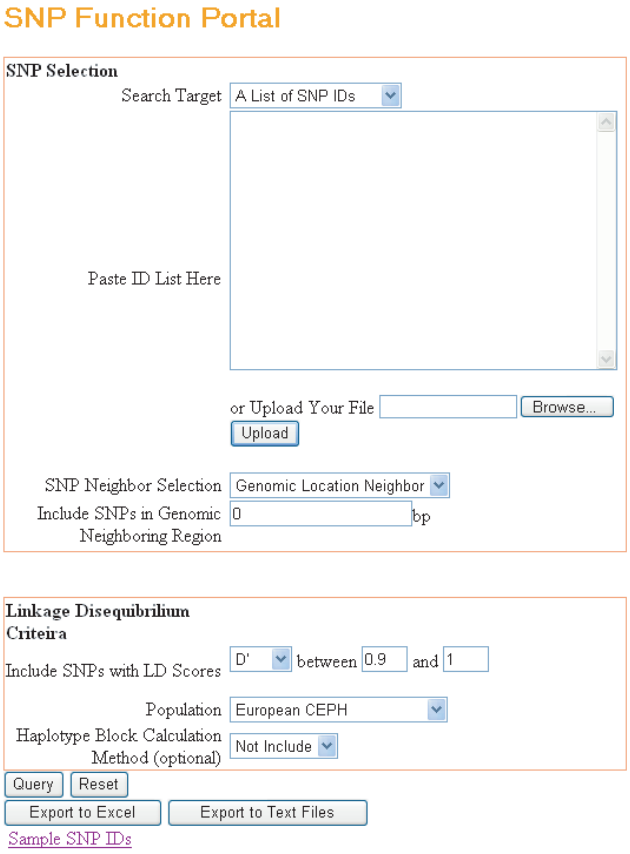


Fig. 3. SNP portal search interface.

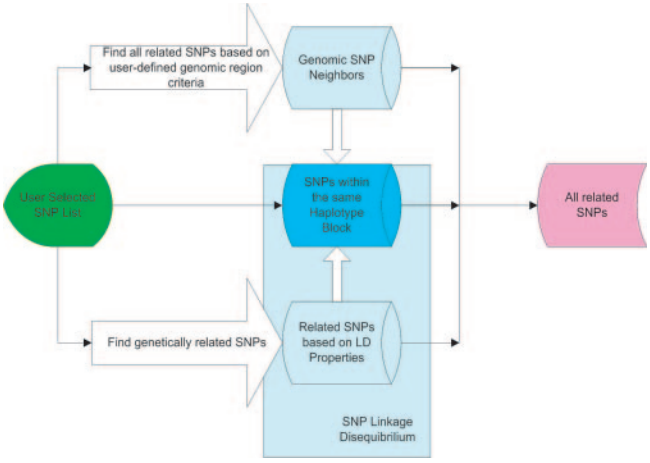


Fig. 4. Data flow of searching related SNPs.

analysis requirements or integrate our SNP search function in other web applications.

Direct download for external sources For all the data we processed and integrated in our database, we provide direct downloads of raw data if researchers want to analyze or further process them in- house. Data can also be downloaded as Microsoft Excel spreadsheets or text files.

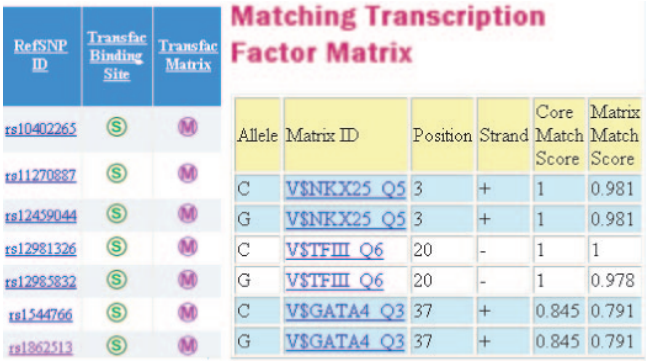


Fig. 5. Transcription Factor Binding Site Annotation. Figure 5 shows the output of transcription factor binding site annotation for a group in the resistin gene. When navigating to the Transcript Level, the user can click the TRANSFAC binding site or matrix next to a SNP ID, and view the potential matches for transcription factor binding sites for the SNP flanking sequences. The transcription factor binding sites shown above are for rs1862513, which was identified in the promoter region of the resistin gene (Banerjee & Lazar, 2003). The SNP variation of rs1862513 is shown to cause different matching scores to the binding site of transcription factor c-Myc (Transfac Matrix ID V$TFIII_Q6).

Web Service: Since we believe our SNP search function utilizing the HapMap II data as well as genomic location of gene and other genetic markers is unique and very powerful, we also establish a web service at http://brainarray.mbni.med.umich.edu/snpservice/service.asmx, for researchers who only want to retrieve the list of related SNPs through our search engine without any annotation attached. The web service will take the same user defined parameters as those in the web query interface in Figure 3. The description of parameter calls is available at http://brainarray.mbni.med.umich.edu/snpservice/SNP_service_desc.htm, and the WSDL description of the web service can be retrieved at http://brainarray.mbni.med.umich.edu/snpservice/service.asmx?WSDL. We will also evaluate options to add new capabilities to the web service and make it compliant to common biological exchange protocol, such as BioMOBY.

3 EXAMPLES

Our SNP Function Portal can be a very convenient tool for researchers to identify and evaluate possible SNP targets for genotyping studies. Complementary to full annotations at the single SNP level provided by the dbSNP and Ensembl, our portal takes a list of SNPs or genetic markers for a comprehensive annotation view together with a SNP LD score filter.

For example, resistin is a peptide hormone produced by adipocytes that may provide a mechanistic link between insulin resistance and obesity (Steppan *et al*., 2001). To investigate the functions of SNPs in the resistin gene, a group of researchers conducted a literature search to identify previously associated SNPs (Conneely *et al*., 2004). The process could be very time consuming and the information found in the literature may also need to be compared with the most recent SNP annotations. Our portal provides a convenient solution to this kind of cherry picking task. With our Search SNP tool, the gene name ''resistin'' identified 21 SNPs using gene-based searching approach. Furthermore, our search tool in SNP Function Portal identified additional 6 SNPs in the same linkage disequilibrium region with *D'* score greater than 0.9, which are about 30kbp upstream of resistin. As a result, researchers can

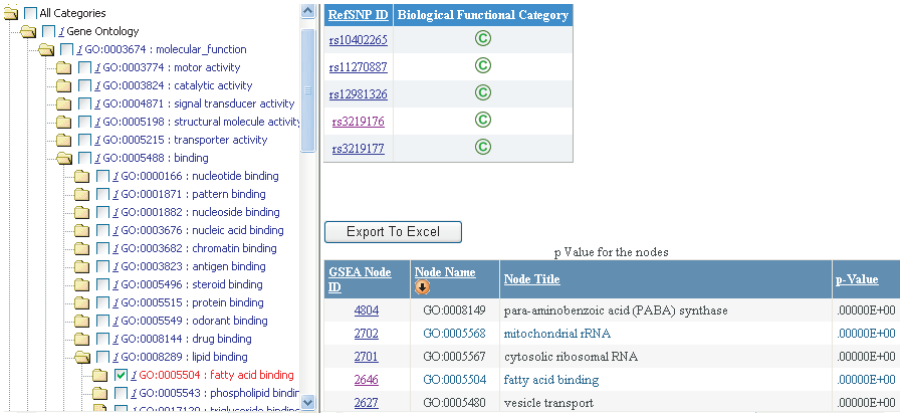


Fig. 6. Gene Ontology Annotations Our Pathway Level features a tree view of common biological functional categories. When users navigate to this level, a click on the ''C'' icon next to a SNP ID will open the category tree on the left pane in the same browser to display all the matched categories in Gene Ontology, Biocarta, KEGG Pathway, etc. for the user selected SNP. In addition, the Fisher's Exact Test p-value for the group of output SNPs matching to functional categories will also be displayed. The example in the figure show that rs3219176 can be mapped to the molecular function of fatty acid binding.

quickly get an overview of all SNPs that may be related to the resistin gene in our system.

In addition, our custom annotations on transcription factor binding sites, protein domain changes and pathway matches can provide researchers additional insights that are not readily available in the current public SNP databases. For example, in our analysis, rs2076489 occurring in the coding region of GABA B receptor may cause domain changes of the 14-element fingerprint, which is the signature of type 2 GABA receptors. Furthermore, our matching of SNP to transcription factor binding matrix suggests that rs2251 may cause changes in binding sites of transcription factor Pit-1 or hypoxia induced factor.

Without doubt, the output is limited by available annotations, and most SNPs cannot be associated with any functional annotation right now. We will continue updating our annotation database to ensure that the SNP Function Portal always contains the most extensive set of annotations available.

4 DISCUSSION

In summary, the SNP Function Portal is a one-stop solution for exploring the potential functional implications of different types of genetic markers through a powerful HapMap II-based search function. Although we certainly need to add more SNP functional annotation data sets, it is already the most powerful SNP annotation web function in the public domain.

In the next phase of development, we plan to consider the use of haplotypes in protein domain analysis. The current version is based on the effect of individual SNPs, as the current HapMap data available at that time did not include enough coding region nonsynonymous SNPs for generating meaningful haplotype-based protein domain analysis data for most of the proteins.

Analyzing the effect of a SNP on gene transcription is a major challenge. Although the TRANSFAC database is the most complete database for transcription factor binding sites, it is far from complete and *de novo* transcription factor binding site identification methods developed in recent years should be helpful (Thompson *et al*., 2003). In addition, even if a SNP is found to change a transcription factor binding site significantly, it is still hard to predict its effect on transcription of the corresponding gene. This is because there is still no accurate model to predict the effect of each transcription factor binding site modification. It can be expected that the accumulation of genotyping data, together with gene expression data from the same samples should provide more reliable data on the influence of SNPs on gene transcription.

Natural language processing approaches for extracting genetic marker-disease relationships in free text databases can also be used to identify relationships between genetic markers and other molecular, cellular and organism-level processes. Such literature derived information will be complementary to the largely sequence-based SNP functional annotation since it is not limited to the molecule/sequence containing the SNP itself but may focus on inter molecular or higher level functions.

We want to point out that the SNP Function Portal is designed for effective SNP function exploration rather than for SNP function prediction. Nonetheless, by incorporating functional annotation data from various data sources and analysis methods, it will become an important tool for understanding genotyping data derived from genome-wide scanning and promote the generation of testable hypotheses. We are committed to updating the SNP Function Portal on monthly basis to keep up with the rapid development in this field.

ACKNOWLEDGEMENTS

P. Wang, M. Dai, W. Xuan, S. J. Watson and F. Meng are members of the Pritzker Neuropsychiatric Disorders Research Consortium, which is supported by the Pritzker Neuropsychiatric Disorders Research Fund L.L.C. This work is also partly supported by the National Center for Integrated Biomedical Informatics through NIH grant 1U54DA021519-01A1 to B.A.

REFERENCES

Banerjee,R. and Lazar,M. (2003) Resistin: molecular history and prognosis. *Journal of Molecular Medicine*, **81**, 218–226.

Bao,L. and Cui,Y. (2005) Prediction of the phenotypic effects of non-synonymous single nucleotide polymorphisms using structural and evolutionary information. *Bioinformatics*, **21**, 2185–2190.

Barrett,J.C., Fry,B., Maller,J. and Daly,M.J. (2005) Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*, **21**, 263–265.

Conneely,K.N., Silander,K., Scott,L.J., Mohlke,K.L., Lazaridis,K.N., Valle,T.T., Tuomilehto,J., Bergman,R.N., Watanabe,R.M., Buchanan,T.A., Collins,F.S. and Boehnke,M. (2004) Variation in the resistin gene is associated with obesity and insulin-related phenotypes in Finnish subjects. *Diabetologia*, **47**, 1782–1788.

Gabriel,S.B., Schaffner,S.F., Nguyen,H., Moore,J.M., Roy,J., Blumenstiel,B., Higgins,J., DeFelice,M., Lochner,A., Faggart,M., Liu-Cordero,S.N., Rotimi,C., Adeyemo,A., Cooper,R., Ward,R., Lander,E.S., Daly,M.J. and Altshuler,D. (2002) The Structure of Haplotype Blocks in the Human Genome. *Science*, **296**, 2225–2229.

Kang,H.J., Choi,K.O., Kim,B.D., Kim,S. and Kim,Y.J. (2005) FESD: a Functional Element SNPs Database in human. *Nucleic Acids Res.*, **33**, D518–522.

Karchin,R., Diekhans,M., Kelly,L., Thomas,D.J., Pieper,U., Eswar,N., Haussler,D. and Sali,A. (2005) LS-SNP: large-scale annotation of coding non-synonymous SNPs based on multiple information sources. *Bioinformatics*, **21**, 2814–2820.

Kasprzyk,A., Keefe,D., Smedley,D., London,D., Spooner,W., Melsopp,C., Hammond,M., Rocca-Serra,P., Cox,T. and Birney,E. (2004) EnsMart: a generic system for fast and flexible access to biological data. *Genome Res.*, **14**, 160–169.

Kel,A.E., Gossling,E., Reuter,I., Cheremushkin,E., Kel-Margoulis,O.V. and Wingender,E. (2003) MATCH^{TM}: a tool for searching transcription factor binding sites in DNA sequences. *Nucl. Acids Res.*, **31**, 3576–3579.

Maglott,D., Ostell,J., Pruitt,K.D. and Tatusova,T. (2005) Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res.*, **33**, D54–58.

Reumers,J., Schymkowitz,J., Ferkinghoff-Borg,J., Stricher,F., Serrano,L. and Rousseau,F. (2005) SNPeffect: a database mapping molecular phenotypic effects of human non-synonymous coding SNPs. *Nucleic Acids Res.*, **33**, D527–532.

Steppan,C.M., Bailey,S.T., Bhat,S., Brown,E.J., Banerjee,R.R., Wright,C.M., Patel,H.R., Ahima,R.S. and Lazar,M.A. (2001) The hormone resistin links obesity to diabetes. **409**, 307–312.

Thompson,W., Rouchka,E.C. and Lawrence,C.E. (2003) Gibbs Recursive Sampler: finding transcription factor binding sites. *Nucleic Acids Res.*, **31**, 3580–3585.

Wang,N., Akey,J.M., Zhang,K., Chakraborty,R. and Jin,L. (2002) Distribution of Recombination Crossovers and the Origin of Haplotype Blocks: The Interplay of Population History, Recombination, and Mutation. *Am. J. Hum. Genet.*, **71**, 1227–1234.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e530–e538
doi:10.1093/bioinformatics/btl208

Protein classification using ontology classification

K. Wolstencroft[1,2,*], P. Lord[1,#], L. Tabernero[2], A. Brass[1,2] and R. Stevens[1]

[1]School of Computer Science and [2]Faculty of Life Sciences, University of Manchester, UK

ABSTRACT

Motivation: The classification of proteins expressed by an organism is an important step in understanding the molecular biology of that organism. Traditionally, this classification has been performed by human experts. Human knowledge can recognise the functional properties that are sufficient to place an individual gene product into a particular protein family group. Automation of this task usually fails to meet the 'gold standard' of the human annotator because of the difficult recognition stage. The growing number of genomes, the rapid changes in knowledge and the central role of classification in the annotation process, however, motivates the need to automate this process.

Results: We capture human understanding of how to recognise members of the protein phosphatases family by domain architecture as an ontology. By describing protein instances in terms of the domains they contain, it is possible to use description logic reasoners and our ontology to assign those proteins to a protein family class.

We have tested our system on classifying the protein phosphatases of the human and *Aspergillus fumigatus* genomes and found that our knowledge-based, automatic classification matches, and sometimes surpasses, that of the human annotators. We have made the classification process fast and reproducible and, where appropriate knowledge is available, the method can potentially be generalised for use with any protein family.

Availability: All components described in this paper are freely available.
OWL ontology http://www.bioinf.man.ac.uk/phosphabase
myGrid http://www.mygrid.org.uk
Instance Store http://instancestore.man.ac.uk

Contact: KWolstencroft@cs.man.ac.uk

INTRODUCTION

Classification of proteins is a central process in understanding the molecular biology of an organism. Sequencing is a first step in revealing the molecular machinery of a cell, but the sequences need to be characterised and classified, at DNA and protein levels, before biologists can start more thorough investigations. Techniques involved in sequencing, especially the high throughput sequencing of whole genomes, have improved dramatically in recent years. Consequently, classification and analysis of data is now the rate-limiting step. This paper describes the addition of an ontology that captures human understanding of recognizing types of protein to the process of automatic classification. By combining this knowledge with existing tools for detecting sequence features we are able to provide a thorough, systematic analysis of a protein family in different genomes, illustrating the utility of such a method in comparative genomics. This methodology does not use any new bioinformatics techniques or algorithms for detecting sequence features. Instead, it augments existing tools by providing a novel method for interpreting the results of these techniques and algorithms to perform automatic protein classification.

Approaches to analysing the large data sets produced in genome sequencing projects have ranged from the 'gold-standard' of human expert annotation to the simple automation of tools such as BLAST (Altschul *et al*., 1997) and Interpro (Mulder *et al*., 2005). Expert analysis enables protein classification to be driven by community knowledge and can add rich, accurate information to data, but it is a time-consuming process and many academic institutions cannot support large teams of bioinformaticians required for such activities. Automated classification methods tend to be quicker, but the level of detail is often reduced, only classifying proteins into broad categories.

Many proteins are assemblies of sequence motifs and domains Each domain or motif might have a separate function within the protein, such as catalysis or regulation, but it is the overall composition that gives each protein its specific function. Recognition of domain and motif composition is a powerful bioinformatics technique which can be employed in the classification of proteins.

There are many tools dedicated to discovering these protein features, including functional domains. For example, PROSITE (Hulo *et al*., 2005), SMART (Letunic *et al*., 2004), and Pfam (Bateman *et al*., 2004) all detect various sequence features. These tools each employ different methods of analysis, for example, PROSITE uses simple pattern-matching to single motifs, whereas Pfam uses hidden markov models (HMMs).

The tool InterproScan encapsulates these, and many other functional domain resources, enabling the use of all from one query submission. In this paper we will refer to protein domains and motifs as p-domains (for protein domains), and we define p-domains as functional units of a protein that have been identified using sequence analysis tools within the InterPro collective.

InterproScan is an efficient automation of p-domain analysis, but while it reports the presence of p-domains, it does not report to which family or subfamily a protein belongs. Bioinformaticians are required to interpret this data in order to classify the protein. In certain cases, the presence of a p-domain is diagnostic for membership of a particular protein family; for example, the protein tyrosine kinase catalytic domain is diagnostic of the tyrosine

*To whom correspondence should be addressed.
#Since moved to Computing Science, University of Newcastle Upon Tyne, UK

kinases. However, classification at a fine-grained level, classifying proteins into subfamilies, is not usually possible without further analysis and interpretation over a collection of revealed sequence features. For automated classification methods, this need for extra human intervention limits performance.

Ontologies provide a technology for capturing and using this human understanding of a domain within computer applications (Stevens *et al*., 2003). In biology, the use of ontologies to capture human knowledge of a particular research area and annotate data is becoming well established. For example, the Gene Ontology describes all gene products common to eukaryotic genomes, promoting common understanding across the community and the MGED ontology provides standardised descriptions of microarray experiments (The Gene Ontology Consortium 2004, Stevens *et al*., 2003). Less well established in the community is the use of reasoning over formal ontologies and their instances, enabling data interpretation. In this study, we present a new method which makes use of this ontological reasoning and illustrates the advantages of such an approach. Our method combines the advantages of human expert analysis and the use of community knowledge with the benefits of increased speed in automated annotation methods. We use a protein family-specific ontology, defined in the OWL language (Dean *et al*., 2004), to capture community knowledge of a protein family together with p-domain analyses, using InterproScan, to automate the characterisation of each protein in that family.

In this paper, we use the protein phosphatase family as a case study. The method we have developed enables the analysis of all protein phosphatases in a genome. To demonstrate its use, we present the analysis of the protein phosphatases of the human and *Aspergillus fumigatus* genomes. We find that in classifying proteins, our system can perform at least as well as a human phosphatase expert. In addition, the systematic and thorough analysis of all protein phosphatases revealed several interesting putative p-domain architectures that were not included in the human expert classifications. We conclude with a discussion of these results and their implication for automatic analysis of genomes.

The protein phosphatase family

Protein phosphatases and protein kinases control phosphorylation events in the cell, which regulate many different aspects of cell life and cell interactions with the environment. Recent reviews on the protein phosphatase family (Alonso *et al*., 2004, Cohen, 1997, Andersen *et al*., 2004) focus on either tyrosine phosphatases or serine/threonine phosphatases. There have been extensive studies into the characterisation of each in the human genome. Although each type of phosphatase performs the same chemical reaction in the cell, the removal of a phosphate group, there are distinct differences in their biological roles and catalytic specificity (Barford, 1996).

Most serine/threonine proteins are multi subunit complexes, combining a catalytic subunit with regulatory and targeting subunits. The final combination of subunits produces the resulting number of each serine/threonine phosphatase in a given organism. For example, the protein phosphatase 1 catalytic subunit binds to different regulatory subunits. Approximately 100 of these regulatory subunits have been identified to date (Bollen, 2001), providing differences in substrate specificity, subcellular localisation and enzymatic activity.

The tyrosine phosphatase family presents a less complicated picture. Instead of protein complexes, they are single polypeptides

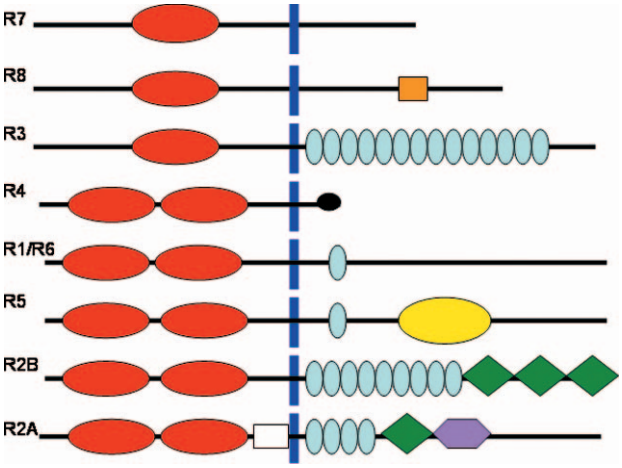


Fig. 1. The differences in domain architecture of the receptor tyrosine phosphatase subfamily. Red = phosphatase catalytic domain. Blue bar = transmembrane region, green = immunoglobulin domain, blue circle = fibronectin domain, purple = MAM domain, yellow = carbonic anhydrase domain, orange = adhesion recognition site, black = glycosylation and white = cadherin-like domain.

with different subtypes providing differences in specificity or subcellular and/or tissue location. However, the necessity for fine-grained classification is increased with the subtlety of the differences between closely related proteins performing different functions. Figure 1 shows the differences in p-domain architecture of the receptor tyrosine phosphatase subfamily of proteins.

The recent implication of phosphatases in human diseases, such as diabetes, cancer and neurodegenerative conditions (Schonthal, 2001, Zhang, 2001 and Tian & Wang, 2002), makes the protein phosphatase family an interesting target for medical and pharmaceutical research and the size of the family means that classification at a detailed level is vital for understanding the biological role of individual proteins and for comparative genomic studies.

Phosphatase ontology

An ontology attempts to describe what exists in the world; an ontology of protein phosphatases describes what protein phosphatases exist. In computer science, an ontology creates a model of what a community understands about its domain as a highly interconnected hierarch of concepts and relationships. By agreeing upon an ontology and the terms within it, a community can create a shared understanding of their domain of study. Committing to such an ontology and its definitions can be used in several ways. One of the most common uses is as a reference; to remove semantic heterogeneity in a community in querying and integration. This has been demonstrated most prominently by the Gene Ontology (Go consortium, 2004), where some 20 databases now use the same terminology to describe the major attributes of functionality of gene products. The GO based descriptions of data have utility not only in retrieval across many resources, but also for analysis of data in, for instance, microarray experiments.

As well as being used as a community knowledge reference, OWL-based ontologies can also be used to perform reasoning. In this work, we utilise the structure and reasoning capabilities of OWL to produce a formal representation of the protein phosphatase

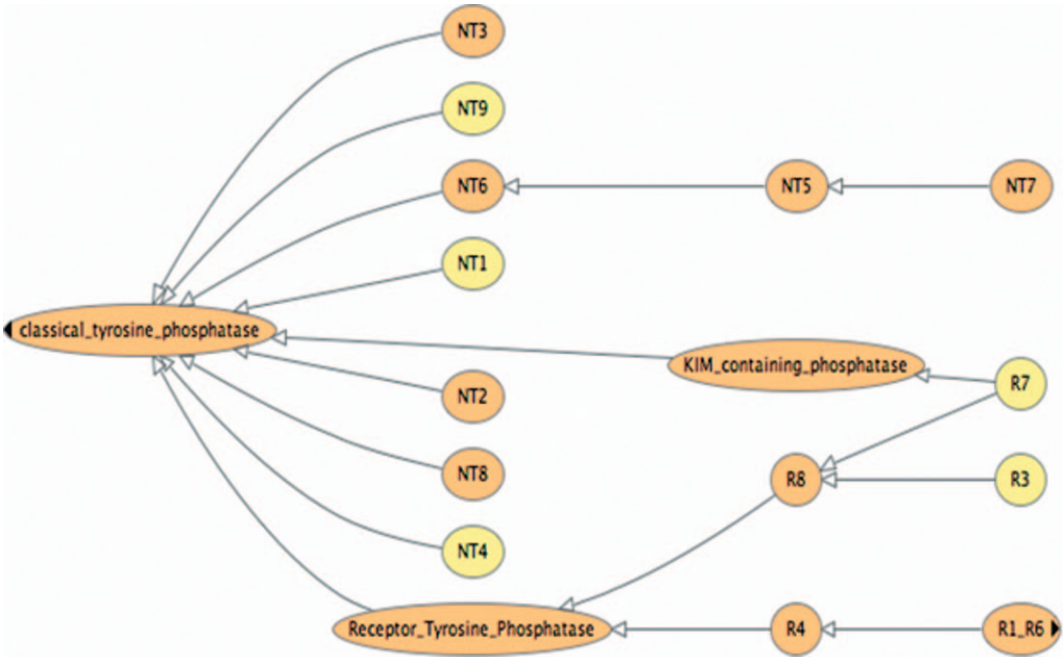


Fig. 2. An OWLViz representation of part of the protein phosphatase ontology.

family classification (derived from the protein phosphatase research community). We then use this classification in order to assign any given individual protein to a particular type of phosphatase based entirely on how the ontology defines a type of phosphatase.

In computer science, an ontology consists of representations of the *things* in the world, often called classes, frames, types or sets, and the relationships between them, often called properties, slots or roles. An OWL ontology contains classes and instances and binary relationships between those instances. Classes represent sets of instances in the world being modeled. In an OWL ontology, when an instance of one class holds a relationship to another, each instance in that class must hold that relationship. So, OWL describes universals. OWL allows precise descriptions of the successors of these relationships. For example, in a class Phosphatase, we describe that *all* instances hold a relationship '*has domain*', which has a successor *Phosphatase Catalytic Domain*. We can say that there must be at least one successor (existential) or that only an instance from *Phosphatase Catalytic Domain* can act as successors (universal). Any relationships held by a class are inherited by all subclasses; the relationships held by instances of a class can only be added to or the successors specialized in their own inheritance trees. As can be seen from Figure 1, this interpretation of world suits the situation found with protein phosphatases. Each member of the family contains a *phosphatase catalytic domain*. Each sub-type simply adds more or different p-domains. This makes it highly amenable to modeling in OWL.

By describing universals an OWL ontology says that all the instances in a class must hold a particular relationship with a particular successor. This is a *necessary* relationship. Additionally, OWL can say that these relationships can be both *necessary* and *sufficient*. This means that when an instance holds such a combination of relationships, then that is *sufficient* to recognize that instance as being a member of that class. In our example, a protein having a protein phosphatase catalytic domain is sufficient to place it into the protein phosphatase class. In this way, OWL ontologies can contain *definitions* of classes in terms of the relationships instances of those classes hold.

The strict and precise semantics of the OWL language mean that it is amenable to automatic reasoning. OWL itself is based upon a decidable fragment of first order logic (Baader, *et al*., 2003), which means it can be submitted to a reasoner. Such a reasoner can determine whether the set of axioms describing the ontology are satisfiable in any world. This practically means that it will report any logical inconsistencies in the ontology. It will also infer the hierarchy of classes implied by the descriptions given in the ontology and thus aid in the creation of a robust classification of types or classes in the ontology.

A protein phosphatase ontology expressed in OWL can capture the necessary and sufficient properties for membership in each protein phosphatase subfamily. For example, in our ontology descriptions of classes, an R5 phosphatase is a type of classical receptor tyrosine phosphatase. As a tyrosine phosphatase, it contains at least one phosphatase catalytic p-domain and as a receptor tyrosine phosphatase, it contains a transmembrane region. From figure 1, it can be seen that this is true for all receptor tyrosine phosphatases. Additionally, the R5 type actually contains two catalytic p-domains and a fibronectin p-domain, placing it into further subclasses. The presence of the distal carbonic anhydrase domain is unique to the R5 type of tyrosine phosphatase. Any protein instance exhibiting all of the above sequence features would be assigned as an instance of the R5 receptor tyrosine phosphatase class.

Figure 2 shows an OWLViz representation of the protein phosphatase p-domain ontology.

By describing protein phosphatases in terms of the p-domains they contain, the phosphatase ontology captures what a human

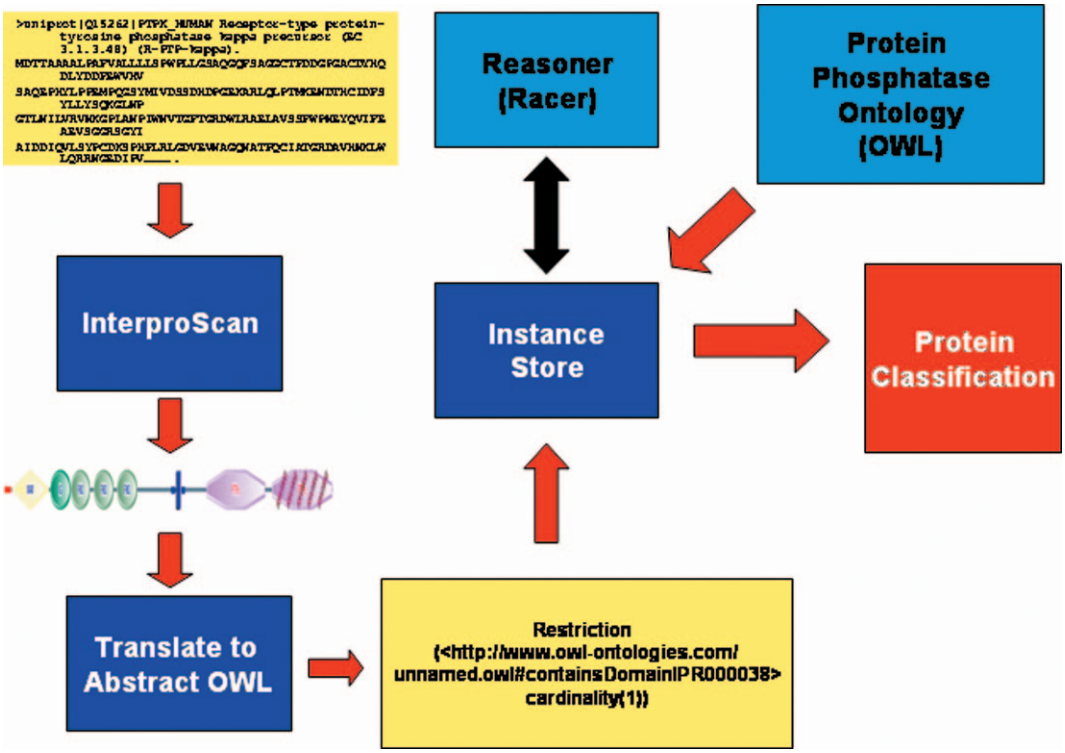


Figure 3. The ontology classification system architecture.

biologist must recognise in an individual protein in order to place it within a type of protein phosphatase. The ontology itself represents a *classification* of phosphatases. OWL ontologies, however, not only represent a classification, but can also perform the act of *classifying*. OWL can not only represent class descriptions, but also do the same for individuals. Given a classification of types or classes, a reasoner can take individuals represented in OWL using the same terms as the ontology and classify them against that ontology. This is how we use the ontology in this protein classification methodology.

The protein phosphatase ontology described in this work only describes the sequence features of proteins, but it was derived from a wider protein family ontology capturing, for example, knowledge about substrates, products, inhibitors and disease associations. This family ontology was built as part of a management system for a protein phosphatase database (Wolstencroft *et al*., 2005). By automatically assigning family and subfamily classifications to new proteins using this domain ontology and reasoning, we hope to infer new knowledge for uncharacterised proteins in the database.

MATERIALS AND METHODS

The bioinformatics analyses necessary for classification of a protein sequence as a protein phosphatase can be divided into the following stages:

(1) Extract the protein phosphatase gene products from the genome in a pre-screening step, without extracting any non-phosphatase proteins

(2) Perform an InterproScan on each protein phosphatase to determine its p-domain composition.

(3) Use the pattern of p-domain composition to identify to which class of phosphatases each protein belongs.

Step three in this analysis usually requires human analysis, but in our method, this is supported computationally by the use of the protein phosphatase ontology. Steps 1 and 2 are already well catered for with bioinformatics tools such as InterPro. As noted in the introduction, it is the stage of using the information provided by such tools that usually requires human intervention. Our ontology, however, captures definitions of what sequence features need to be present in an individual protein for it to be recognized as a particular type of protein phosphatase. The essence of our methodology is to use this ontology within an application to recognize the consequences of p-domain detection by InterPro for membership in a particular class of protein phosphatases.

The ontology describes classes of phosphatases, but not individual proteins. In order to reason over the descriptions of individual proteins, as described in the previous section, we use a related technology, the Instance Store (Horrocks *et al*., 2004).

The Instance Store combines a Description Logic reasoner with a relational database. The reasoner in this case performs the task of classification; that is, from the OWL instance descriptions given, it determines the appropriate ontology class for an instance description. The relational database provides the stability, scalability and persistence necessary for this work. The Instance Store itself provides a relatively simple programmatic interface, allowing assertion of descriptions and queries against the set of instances. It uses highly optimized algorithms to denormalise datasets as they are asserted and later determine whether the information in the database is sufficient to answer queries, or whether reasoning is required.

The automated classification system we have developed combines elements from the myGrid service-orientated architecture described previously (Stevens *et al*., 2004) with description logic reasoning (Baader *et al*., 2003) to extract and classify the protein phosphatase gene products from an organism. The system uses the OWL protein phosphatase ontology; the

Instance Store and a reasoner to classify the individual proteins. Figure 3 shows the architecture of the system. The use of myGrid services allows large data sets to be passed through many stages of analyses without the need for human intervention and without the need for the installation and maintenance of local databases and bioinformatics tools (Oinn *et al*., 2004).

The data sets

The study focuses on the human and the *Aspergillus fumigatus* protein phosphatases. The human phosphatases had already been identified and extensively described in previous studies (Alonso *et al*., 2004, Mustelin *et al*., 2005), but in Aspergillus, the protein phosphatases required identification and extraction from the genome.

Previous classification of human phosphatases by biological experts provides a substantial test-set for the ontology. If the ontology classifies the proteins as well as the human experts have, studies on new, unknown genomes can be undertaken with greater confidence. The *Aspergillus fumigatus* genome offers a unique insight into the comparison between the automated method and the manual. The *A. fumigatus* genome has been sequenced and annotation is currently underway by a team of human experts (Mabey *et al*., 2004).

Pre-screening The proteome datasets were pre-screened to isolate phosphatase proteins from the rest of the organism's proteins. This was achieved by screening for diagnostic phosphatase p-domains. These are: a) the protein tyrosine phosphatase active site motif H-C-X(5)-R (Andersen *et al*., 2005) b) the protein serine/threonine phosphatase motif [LIVMN]-[KR]-G-N-H-E and c) the protein phosphatase C signature motif [LIVMFY]-[LIVMFYA]-[GSAC]-[LIVM]-[FYC]-D-G-H-[GAV]. The EMBOSS program patmatdb (Rice *et al*., 2000) was used for this initial screening process.

This pre-screening process is not strictly necessary. Performing an InterProScan on each and every protein would enable the isolation of phosphatase proteins, but this step is time-consuming. One InterproScan can take up to ten minutes to perform, whereas, patmatdb can screen the whole dataset in less than a minute, reducing the overall experiment time.

InterPro

Proteins identified in the pre-screen were individually searched against Interpro using InterproScan as a web service implementation provided by the EBI. Results were gathered in XML, which was then parsed into a tab delimited format containing the Interpro accession number(s) and the numbers of times p-domains occurred.

Translation into OWL

The translation into OWL instance descriptions is largely a syntactic transformation from the previous step, although it requires implicit knowledge of the ontology. In this case, the use of naming conventions within the ontology made this transformation simple. Once translated, all descriptions of protein instances were loaded into the Instance Store. We then systematically asked the instance store which proteins belonged to which class of phosphatase. The end result of this processing was a report on the numbers and types of protein phosphatases in a genome.

RESULTS

In order to demonstrate the performance of the ontology driven classification system, the proteins identified and classified in previous human phosphatase reviews (Alonso *et al*., 2004, Cohen, 1997, and Andersen *et al*., 2004) were used to compare the ontology classification to that derived by human experts. Table 1 shows the number of proteins in each of the higher level protein phosphatase subfamily classes in the human classification and in the automated classification.

The comparison between the classifications clearly demonstrates that the performance of the automated ontology classification system is equal to that of the human annotated original. The ontology class definitions were sufficient to identify the differences between protein subfamilies and demonstrate the usability of the system on uncharacterised genomes.

Table 1. A comparison of the numbers of proteins assigned to subfamilies of protein phosphatases by expert annotation and by the automated ontology classification. The numbers of protein instances in the serine/threonine subfamilies looks much smaller than the PTPs. This is because the PTPs are single subunit proteins, whereas the serine/threonine PPPs are predominantly multi-subunit proteins. The instances presented are the catalytic subunits alone. Modeling the regulatory subunits in a similar manner was out of the scope of this project, but would be an interesting extension.

Phosphatase classification	Human expert classification	Ontology classification
Tyrosine Phosphatases		
A Class I Cys-based PTPs	**93**	**93**
Classical PTPs	36	36
Receptor type	19	19
Non-receptor type	17	17
VH1-like (DSPs)	57	57
MKP	11	11
Myotubularins	14	14
'atyipical' DSPs	17	17
B Class II Cys-based PTPs	1	1
Low molecular weight PTPs		
C Class III Cys-based PTPs		
CDC25	3	3
D Asp-based PTPs		
EyA	4	4
Serine/threonine Phospahtases		
PPP		
Classical		
Novel	10	10
PPP5	1	1
PPP7	1	1
PPM	5	5
PP2C		

Table 1 illustrates protein numbers in phosphatase subfamilies. However, many subfamilies contain several different subtypes. For example, there are eight subtypes of receptor tyrosine phosphatases and seven subtypes of myotubularins.

An interesting result from the analysis was that, using the ontology, we were able to identify additional functional domains in two dual specificity phosphatases, presenting the opportunity to refine the classification of the subfamily into further subtypes.

Alonso *et al*. (2004), describe the 'atypical' dual specificity phosphatases as being divided into seven subtypes. The largest of these have the same p-domain architecture; they contain tyrosine phosphatase and dual specificity catalytic p-domains alone. However, several proteins have additional functional domains that have been shown to confer functional specificity (Wang *et al*., 2001). Classifying the proteins using the ontology highlighted more of these 'extra' p-domains.

The protein DUS12 contains a zinc finger domain (IPR007087). This protein has been characterised not only in the human genome (Marco *et al*., 1999), but in many other species (Kumar *et al*., 2004).

Table 2. *A.fumigatus* protein phosphatases classified by the automated ontology system

Phosphatase classification	Number of proteins	Protein identifiers
Tyrosine Phosphatases		
A Class I Cys-based PTPs		
Classical PTPs		
Non-receptor type	3	Afu3g10970
		Afu4g07000
		Afu6g06650
Receptor type	0	
VH1-like (DSPs)		
MKP	1	Afu4g04710
Myotubularins	1	Afu1g05640
'atypical' DSPs	6	Afu2g11990
		Afu2g02760
		Afu3g12250
		Afu5g11690
		Afu4g07080
		Afu1g03540
B Class II Cys-based PTPs		
Low molecular weight PTPs	0	
C Class III Cys-based PTPs		
CDC25	0	
D Asp-based PTPs		
EyA	0	
Serine/threonine Phospahtases		
PPP		
Classical	9	Afu2g03950
		Afu5g12010
		Afu5g11370
		Afu5g09360
		Afu5g08620
		Afu5g06700
		Afu1g04950
		Afu6g10830
		Afu6g11470
Novel		
PPP5	1	Afu5g06700
PPP7		
PPM		Afu1g15800
PP2C	5	Afu1g09280
		Afu2g03890
		Afu5g13340
		Afu5g13740

In the classification presented by Alonso *et al*., (2004) the protein is present, but is miss-annotated as containing a FYVE domain. FYVE domains are different types of zinc finger domains which occur in the myotubularin proteins MTMR3 and MTMR4. Earlier reviews of the tyrosine phosphatase family, however, do include the zinc finger domain in the protein (Bhaduri & Sowdhamini, 2003). These results illustrate an inconsistency in the accepted protein phosphatase community knowledge and highlight a possible disadvantage of human expert annotation, namely human error leading to omission.

The dual specificity phosphatase 10 protein (DUSP10) contains a disintegrin domain. The UniProt record reflects this, but the domain does not appear in any phosphatase characterisation/classification studies. The domain architecture of DUSP10 is conserved in other species (data not shown), which suggests a specific function for the domain, but current experimental evidence does not explain what this might be.

Aspergillus fumigatus

The success of the ontology system in classifying the known human phosphatases enables the classification of phosphatases from incomplete or unannotated genomes. The *A.fumigatus* genome has been partially annotated. It has been sequenced, and is being annotated by human experts. Therefore, the protein data currently consists of both predicted and known proteins. The predicted proteins may contain descriptions based upon automated similarity searches, producing entries termed 'hypothetical' or 'putative', but their annotation is limited.

Using the ontology system to classify the phosphatases allows a comparison between the proteins already annotated and those with partial annotation from similarity searching. Table 2 summarises the classes of *A.fumigatus* protein phosphatases identified by the ontology system.

The table illustrates important differences between the phosphatases of the two test organisms. The protein serine/threonine phosphatase composition remains relatively unchanged, but there are radical differences between the tyrosine and dual specificity subfamilies. Firstly, the number of proteins in *A.fumigatus* is greatly reduced. Where the human genome contains 16 myotubularin proteins and 11 MAP kinase phosphatase proteins, *A.fumigatus* contains only one of each. The number of 'classical' protein tyrosine phosphatases is also reduced. There are no incidences of receptor tyrosine phosphatases and only three non-receptor tyrosine phosphatases. These results may initially seem surprising, but the complexity of the two organisms is radically different. Requirements for tissue specificity, for example, are reduced in *A.fumigatus*, and some tyrosine phosphatases have been shown to exhibit tissue-specific expression (Chagnon *et al*., 2004). There is also the issue of the pathways that the 'missing' phosphatases are involved in. Some phosphorylation pathways would be expected to be conserved, but it should also be expected that specific mammalian and fungal pathways would require different phosphatase components.

The ontology classification uncovered a protein phosphatase with a novel domain architecture. Protein Afu5g09360 is a calcineurin protein (PP2B) which contains an extra homeobox domain. The homeobox domain binds to DNA using a helix-turn-helix structural motif. It is found in a variety of DNA-binding proteins, many of which are transcription factors.

PP2B is well conserved throughout evolution. Performing BLAST analyses on Afu5g09360 and InterproScans of the proteins exhibiting the most similarity (data not shown) revealed that the homeobox domain in PP2B was present in other aspergillus species and closely related fungi, but was not present in any other taxa. The conservation strongly suggests a specific function for this extra domain. Previous studies have identified a divergence in the mechanisms of action of calcineurin in pathogenic fungi (Kraus &

Table 3. A comparison of the differences in classification between the annotations assigned to phosphatases by the *A.fumigatus* sequencing project and by the ontology

	A.fumigatus annotation	Ontology classification
Afu1g03540	Hypothetical protein	Dual specificity phosphatase
Afu1g05640	Protein phosphatase	Myotubularin
Afu5g11690	Related to protein tyrosine phosphatase PPS1	Dual specificity phosphatase
Afu4g07080	Putative dual specificity phosphatase	Dual specificity phosphatase
Afu4g07000	Tyrosine phosphatase	Tyrosine phosphatase
Afu4g04710	Putative tyrosine phosphatase	MAP Kinase Phosphatase (MKP)
Afu6g06650	Conserved hypothetical protein	Tyrosine phosphatase
Afu2g11990	Pten-3-phosphoinisitide phosphatase	Dual specificity phosphatase
Afu3g12250	Putative protein tyrosine phosphatase	Dual specificity phosphatase
Afu2g02760	Putative protein tyrosine phosphatase	Dual specificity phosphatase
Afu3g10970	Protein tyrosine phosphatase	Protein tyrosine phosphatase
Afu1g04950	serine/threonine protein phosphatase 1	Classical serine/threonine phosphatase
Afu1g09280	protein phosphatase 2C, putative	Protein phosphatase 2C
Afu1g15800	protein phosphatase 2C, putative	Protein phosphatase 2C
Afu2g03890	Protein phosphatase 2C, putative	Protein phosphatase 2C
Afu2g03950	serine/threonine protein phosphatase, putative	Classical serine/threonine phosphatase
Afu5g06700	serine/threonine protein phosphatase PPT1	Protein phosphatase 5
Afu5g08620	Ser/Thr protein phosphatase family	Classical serine/threonine phosphatase
Afu6g11470	TOR signalling pathway phosphatase, putative	Classical serine/threonine phosphatase
Afu6g10830	protein phosphatase 2a	Classical serine/threonine phosphatase
Afu5g13340	protein phosphatase 2C, putative	Protein phosphatase 2C
Afu5g12010	serine/threonine phosphatase	Classical serine/threonine phosphatase
Afu5g11370	Ser/Thr protein phosphatase	Classical serine/threonine phosphatase
Afu5g09360	calcineurin A	Classical serine/threonine phosphatase—with unique homeobox domain
Afu5g13740	phosphatase 2C, putative	Protein phosphatase 2C

Heitman, 2003) and have also demonstrated that this is critical for virulence. Other studies on one function of calcineurin in Arabidopsis, ion homeostasis, (Shin *et al*., 2004) have revealed a homeobox protein, Athb-12, is also involved. This study raises the possibility of a similar regulatory role for the homeobox domain in the *A.fumigatus* protein, but confirmatory experimental evidence will have to be obtained.

The ontology system vs. *A.fumigatus* genome automated annotation pipeline

Many of the protein phosphatases identified in the ontology classification system had not been classified and curated manually by the *A.fumigatus* genome group, but had simply been annotated using the results of automated annotation methods (Allen *et al*., 2004). In many cases, the automated annotation approach underperformed when compared to the ontology system. The ontology classification placed proteins into more specific classes than the automated approach adopted by the *A.fumigatus* genome group. For example, the ontology classified the protein Afu1g05640 as a myotubularin, a specific subclass of the dual-specificity phosphatases, which is a lipid phosphatase. The annotation from the *A.fumigatus* sequencing consortium simply stated that it was a protein phosphatase. In one case, the *A.fumigatus* annotation appeared to provide a more detailed classification than the ontology. The protein Afu2g11990 was annotated as a Pten phosphatase, whereas the ontology simply classified it as a dual specificity phosphatase (the parent class of Pten). However, on closer inspection, the protein did not contain p-domains indicative of Pten proteins (Alonso *et al*., 2004). A sequence similarity search revealed partial similarity to the Pten protein from *Dictyostelium discoideum*, but this was in the region of the dual specificity phosphatase domain, so there does not appear to be sufficient evidence to place this protein in the Pten phosphatase class.

Table 3 shows the comparative classifications of protein phosphatases in the ontology system and in the automated *A.fumigatus* annotation pipeline.

DISCUSSION

Post-genomic bioinformatics presents new problems for the bioinformatician. The scale of data production has increased dramatically while the pace of data analysis and annotation has not kept pace. Often, compromises on the quality of annotation have to be made in order to interpret large data sets quickly. We have tried to avoid making such a compromise by designing a system that will allow rapid, automated classification to the fine-grained, subfamily level. This study demonstrates the advantages of combining community knowledge, in the form of an ontology, with automated annotation methods.

Standard automated methods of annotation provide evidence for similarity to other known proteins, or provide lists of functional domains within a protein, but they do not allow the interpretation of this information. The strength of human expert annotation is in

this interpretation step. In our novel approach, we were able to replace this interpretation step with further automation. Using the technologies of formal description logics and ontological reasoning, we could capture and utilise community knowledge for data analysis.

By using InterproScan to perform the domain composition analyses, we are able to benefit from the combined advantages of all of the different domain/motif searching techniques developed by the protein domain databases that contribute to Interpro. Our method does not replace the need to use these domain identification tools, nor does it introduce a novel detection method; it simply provides a mechanism for automatically interpreting the results of these searches.

The ontology system classified the human protein phosphatases with equal competence to human experts, enabling confidence to be placed in similar studies of the proteins of uncharacterised genomes. It was also discovered that the ontology system was efficient at uncovering novel, unexpected functional domains and therefore uncovering interesting new targets for future research. The computational use of human knowledge in our methodology allows a systematic, thorough approach to the classification of proteins in a genome. It is possible for a human bioinformatician to perform the same task, but human annotators often have a particular question in mind when searching and consequently may overlook outliers that do not match this pre-formed template. In addition, this system avoids human frailties of slips, omissions and boredom.

The ontology definitions were constructed from what was known to be present. If a domain was found in a protein that did not appear in the ontology, there was a notable inconsistency in the Instance Store, enabling easy identification. In the human study, two of these unexpected domains were identified. The zinc finger domain in the dual specificity phosphatase C protein has been well characterised, first in *Plasmodium falciparum*, and later in other organisms. It is omitted from the most recent phosphatase classification (Alonso *et al*., 2004), but is included in previous works (Bhaduri & Sowdhamini, 2003), which highlights inconsistencies and discrepancies within the phosphatase community knowledge base.

The disintegrin domain identified in DUSP10 provides a more interesting and open biological question. It is a distinct functional domain and is conserved in the DUSP10 protein from other species (data not shown). This conservation suggests a specific role for this domain, but, to date, there is no experimental evidence. In vivo studies on the protein have identified a role in the innate and adaptive immune response and it has also been found to block the enzymatic activity of the MAP Kinases, p38, JNK and SAPK.

The results from *A.fumigatus* also produced interesting biological questions. The homeobox domain identified in protein Afu5g09360 appears to be conserved across Aspergillus species and closely related fungi, but does not appear in any other taxa. This could perhaps suggest a fungal-specific pathway for the phosphatase. A broader question arising from the *A.fumigatus* study is a comparative genomics question. A comparative study of other fungal species and species from other taxa, could greatly increase our understanding of the evolution of protein phosphorylation, and the ontology system developed in this study provides a unique opportunity to gather the data for such a study.

Work with the ontology system can also be expanded to other protein families. Protein phosphatases provided a good use-case and proof of concept for our method, but the method is not confined to one family of proteins. Work is underway to construct a similar ontology for the ABC transporters and potassium channel proteins and eventually we would like to do the same for the protein kinase domain, allowing the extraction and classification of the phosphorylome from new genomes.

The development of such ontologies is limited by our knowledge of the features that determine particular functionality in a protein and the availability of tools to detect all those features. We have observed, for instance, that some classifications are based upon tissue specificity of a protein that is based upon regulation of sequence identical proteins by other genetic features. Detecting such information lies outside the tools currently used in our methodology. In other cases, the ordering of sequence features is important for recognizing protein family type. Such ordering is not usually possible in OWL, but an ontology design pattern now exists for expressing lists and we expect to employ this pattern in the near future. Nevertheless, the possibilities and limitations of our approach remain to be fully explored.

By combining ontology reasoning with the myGrid service layer, we have produced an automated annotation system that can perform genome-wide surveys and protein classification equal to the 'gold standard' of human expert annotation. We believe that this work could also have wider implications within bioinformatics. Currently, the use of ontological technology has been largely restricted to enhancing browsing and querying over existing data. In this paper, we have described the application of the computationally amenable semantics of an OWL ontology to the enhancement of community-developed knowledge. By encoding pre-existing community knowledge in this way, we have gained the advantage of automation and the ability to systematically analyse large volumes of biological data. In this case, this has resulted in the uncovering of interesting biological observations that will lead to further experimental investigation.

While in this paper we have focused on proteins, this method is applicable to any area of biology where properties defining class membership can be derived from automated analysis tools. For these reasons, we believe that this style of automatic classification could have a great impact in bioinformatics analyses.

ACKNOWLEDGEMENTS

This work was funded by an MRC PhD studentship and myGrid e-science project, University of Manchester with the UK e-science programme EPSRC grants GR/R67743 and BBS/B/1713. Preliminary sequence data was obtained from The Institute for Genomic Research website at http://www.tigr.org from Dr Jane Mabey-Gilsenan. Sequencing of *Aspergillus fumigatus* was funded by the National Institute of Allergy and Infectious Disease U01 AI 48830 to David Denning and William Nierman, the Wellcome Trust, and Fondo de Investicagiones Sanitarias

REFERENCES

Allen,J.E., Pertea,M. and Salzberg,S.L. (2004) Computational gene prediction using multiple sources of evidence.. *Genome Res*., **14(1)**, 142–8.

Alonso,A., Sasin,J., Bottini,N., Friedberg,I., Friedberg,I., Osterman,A., Godzik,A., Hunter,T., Dixon,J. and Mustelin,T. (2004) Protein tyrosine phosphatases in the human genome. *Cell*, **117(6)**, 699–711, Review.

Altschul,S.F., Madden,T.L., Schäffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Andersen,J.N., Jansen,P.G., Echwald,S.M., Mortensen,O.H., Fukada,T., Del Vecchio,R., Tonks,N.K. and Moller,N.P. (2004) A genomic perspective on protein tyrosine phosphatases: gene structure, pseudogenes, and genetic disease linkage. *FASEB J*., **18(1)**, 8–30, Review.

Andersen,J.N., Del Vecchio,R.L., Kannan,N., Gergel,J., Neuwald,A.F. and Tonks,N.K. (2005) Computational analysis of protein tyrosine phosphatases: practical guide to bioinformatics and data resources. *Methods*, **35(1)**, 90–114, Review.

Baader,F., Calvanese,D., McGuinness,D., Nardi,D. and Pater-Schneider,P. (2003) *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Barford,D., Das,A.K. and Egloff,M.P. (1998) The structure and mechanism of protein phosphatases: insights into catalysis and regulation. *Annu. Rev. Biophys Biomol. Struct*., **27**, 133–64, Review.

Bhaduri,A. and Sowdhamini,R. (2003) A genome-wide survey of human tyrosine phosphatases. *Protein Eng*, **16(12)**, 881–8.

Chagnon,M.J., Uetani,N. and Tremblay,M.L. (2004) Functional significance of the LAR receptor protein tyrosine phosphatase family in development and diseases. *Biochem. Cell Biol*., **82(6)**, 664–75.

Cohen,P. (1997), Novel protein serine/threonine phosphatases: variety is the spice of life. *Trends Biochem. Sci*., **22(7)**, 245–51, Review.

Horrocks,L., Li,D. Turi and Bechhofer,S. (2004) The Instance Store: DL reasoning with large numbers of individuals. In Proc. of the 2004. *Description Logic Workshop*, pages 31–40.

Hulo,N., Sigrist,C.J.A., Le Saux,V., Langendijk-Genevaux,P.S., Bordoli,L., Gattiker,A., De Castro,E., Bucher,P. and Bairoch,A. (2004) Recent improvements to the PROSITE database. *Nucleic Acids Res*., **32**, D134–D137.

Kraus,P.R. and Heitman,J. (2003) Coping with Stress: Calmodulin and Calcineurin in Model and Pathogenic Fungi Biochemical and Biophysical Research Communications. **311**, 1151–1157.

Kumar,R., Musiyenko,A., Cioffi,E., Oldenburg,A., Adams,B., Bitko,V., Krishna,S.S. and Barik,S. (2004) A zinc-binding dual-specificity YVH1 phosphatase in the malaria parasite, Plasmodium falciparum, and its interaction with the nuclear protein, pescadillo,. *Mol Biochem Parasitol*., **133(2)**, 297–310.

Letunic *et al*. (2004) SMART 4.0: towards genomic data integration. *Nucleic Acids Res*., 32.

Mabey,J.E., Anderson,M.J., Giles,P.F., Miller,C.J., Attwood,T.K., Paton,N.W., Bornberg-Bauer,E., Robson,G.D., Oliver,S.G. and Denning,D.W. (2004) CADRE: the Central Aspergillus Data Repository. *Nucleic Acids Res*., **1(32)**, D401–5.

Mulder,N.J., Apweiler,R., Attwood,T.K. *et al*. (2005) InterPro, progress and status in 2005. *Nucleic Acids Res*., **33**, Database Issue: D201–5.

Oinn,T., Addis,M., Ferris,J., Marvin,D., Senger,M., Greenwood,M., Carver,T., Glover,K., Pocock,M.R., Wipat,A. and Li,P. (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, **22;20(17)**, 3045–54.

OWL Web Ontology Language Reference http://www.w3.org/TR/owl-ref/

Pillutla,R.C., Shimamoto,A., Furuichi,Y. and Shatkin,A.J. (1998) Human mRNA capping enzyme (RNGTT), cap methyltransferase (RNMT) map to 6q16 and 18p11.22-p11.23, respectively. *Genomics*, **154(2)**, 351–3.

Rice,P., Longden,I. and Bleasby,A. (2000) EMBOSS: The European Molecular Biology Open Software Suite. *Trends in Genetics*, **16(6)**,276–277.

Schonthal,A.H. (2001) Role of serine/threonine protein phosphatase 2A in cancer. *Cancer Lett*., **170(1)**,1–13.

Senger,M., Rice,P. and Oinn,T. Soaplab—a unified Sesame door to analysis tools Proceedings. *UK* e-Science, All Hands Meeting Editor—Simon J Cox, pp. 509–513.

Stevens,R., Wroe,W., Lord,P. and Goble,C. (2003) Ontologies in bioinformatics. In Stefan Staab and Rudi Studer, editors, Handbook on Ontologies in Information Systems. pp. 635–657, Springer.

Stevens,R., Tipney,H.J., Wroe,C., Oinn,T., Senger,M., Lord,P., Goble,C.A., Brass,A. and Tassabehji,M. (2004) Exploring Williams-Beuren Syndrome Using myGrid in Proceedings of 12th International Conference on Intelligent Systems in Molecular Biology, 31st Jul-4th Aug 2004, Glasgow, UK, published. *Bioinformatics*, **20** Suppl., i303–i310.

Tian,Q. and Wang,J. (2002) Role of serine/threonine protein phosphatase in Alzheimer's disease. *Neurosignals*, **11(5)**,262–269.

The Gene Ontology Consortium (2004). The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res*., **32**, D258–D261.

Wang,J., Stuckey,J.A., Wishart,M.J. and Dixon,J.E. (2001) A unique carbohydrate binding domain targets the lafora disease phosphatase to glycogen. *J. Biol. Chem*., **2002;77(4)**, 2377–80.

Wolstencroft,K., McEntire,R., Stevens,R., Tabernero,L. and Brass,A. (2005) Constructing ontology-driven protein family databases. *Bioinformatics*, **15;21(8)**, 1685–92.

Zhang,Z.Y. (2001) Protein tyrosine phosphatases: prospects for therapeutics. *Curr. Opin. Chem. Biol*, **5(4)**, 416–23.

Alex Bateman, Lachlan Coin, Richard Durbin, Robert,D., Finn Volker Hollich, Sam Griffiths-Jones, Ajay Khanna, Mhairi Marshall, Simon Moxon, Erik,L.L., Sonnhammer, David,J., Studholme, Corin Yeats, Sean,R. and Eddy (2004) The Pfam Protein Families. *Database Nucleic Acids Res*., **32**, Database Issue: D138–D141.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e539–e546
doi:10.1093/bioinformatics/btl204

Inferring Functional Pathways from Multi-Perturbation Data

Nir Yosef[1,*,†], Alon Kaufman[2,†] and Eytan Ruppin[1,3]

[1]School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel, [2]Center of Neural Computation, Hebrew University, Jerusalem, Israel and [3]School of Medicine, Tel-Aviv University, Tel-Aviv, Israel

ABSTRACT

Background: Recently, a conceptually new approach for analyzing gene networks, the Functional Influence Network (FIN) was presented. The FIN approach uses the measured performance of a given cellular function under different multi-perturbations, to identify the main functional pathways and interactions underlying its processing. Here we present and study an iterative, extended version of FIN, the Functional Influence Network Extractor (FINE), which is specifically geared towards the accurate analysis of sparse cellular systems. We employ it to study a conceptually fundamental question of practical importance—how well should we know the system studied (such that we can predict its performance) so that we can understand its workings (*i.e.,* chart its underlying functional network)?

Results and Conclusions: The performance of FINE is studied in both simulated and biological sparse systems. It successfully obtains an accurate and compact description of the underlying functional network even with limited data, and outperforms FIN. We show that prior estimates of a system's functional complexity are instrumental in determining how much predictive knowledge is required to accurately chart its underlying functional network.

Availability: The FINE software is available for download at http://www.cns.tau.ac.il/resc.html

Contact: niryosef@post.tau.ac.il

INTRODUCTION

Which elements within a system are important for its performance? How do these elements influence the system's performance, and to what extent? Are there inter-element interactions which significantly affect the system's performance? These fundamental questions typically arise when attempting to analyze a system in order to understand its workings. Specifically, within the context of genetic networks, the success of genome sequencing projects and high throughput gene expression studies has allowed biologists to identify almost all genes responsible for producing the biological complexity of several organisms. The next important task is to quantify their importance to various cellular functions (Carpenter *et al*., 2004) and understand their functional regulatory interactions (Barabasi *et al*., 2004) and the 'logic circuitry' (Davidson *et al*., 2002).

To causally deduce the roles played by genes in determining a cellular function, or more generally the role of elements in any system, perturbation studies are necessary and have been traditionally employed. In perturbation studies, phenotypic variation is traced after deletion or mutation of different genes. Nevertheless, the vast majority of these studies have employed single perturbations, which often result in little phenotypic effect, due to the existence of duplicates, alternative pathways and functional overlap (Gu *et al*., 2003). Hence, multiple concomitant perturbations should be employed in order to identify the causal contributions of the different genes to the system's functioning (Kaufman *et al*., 2005). Such studies have been quite scarce up until now, comprised of either large-scale studies of double knockouts (Tong *et al*., 2004), or small-scale studies spanning a broader span of multiple-knockouts (Yuh *et al*., 2001; Kaufman *et al*., 2004). However, the growing awareness that multi-perturbation studies are essential for deciphering the workings of complex genetic networks, along with the recent development of new experimental methods such as RNA interference (RNAi) (Hammond *et al*., 2001) and transposon mutagenesis, will soon lead to the accumulation of large amounts of multi-perturbation genetic data.

The goal of the algorithm at the basis of this paper is to reveal the main functional pathways and functional interactions uncovered by multiple knockout experiments in a genetic network. Obviously, there have been many studies which have developed methods to uncover the network of interactions between genes, mostly based on microarray data. Such studies typically address microarray data analysis by inferring regulatory networks using Boolean networks (Ideker *et al*., 2000), Bayesian networks (Pe'er *et al*., 2001) and other approaches (Ideker *et al*., 2001; Tegner *et al*., 2003). Unlike these approaches, our goal is to obtain causal functional descriptions, by analyzing data gathered in studies where a specific cellular function is probed using a variety of multiple knockout experiments. The functional interactions and pathways we aim to reveal do not necessarily imply any physical or direct biochemical interactions, and rather represent functional modules. Keinan *et al*. (2004) and Kaufman *et al*. (2005) have previously presented two complementary methods to address this challenge, and applied them to the analysis of genetic and neuronal multi-perturbation data. The latter presented the Functional Influence Network (FIN) algorithm, aiming to produce a *Compact Functional Network (CFN)* which describes in a compact and accurate manner how the genes, acting together in functional pathways, determine a certain cellular function or phenotypic behavior. In this study we expand the basic FIN approach in three fundamental ways:

(1) First, we develop a new algorithm—the Functional Influence Network Extractor (FINE), motivated by the empirical observation that many biological networks are functionally sparse

*To whom correspondence should be addressed.
†These authors made an equal contribution to this work.

(*e.g*. Thieffry *et al*., (1998); Jeong *et al*., (2000)), *i.e*. have a compact *functional backbone*.

(2) Second, we perform an extensive study of the workings of FINE. To this end, a comprehensive set of measures was developed to evaluate the performance of the algorithm, on a large number of simulated networks. We then applied FINE to the analysis of the *cis*-regulatory system of the sea urchin *endo16* gene (Yuh *et al*., 2001).

(3) The third contribution that this paper makes is conceptual: since obviously one cannot expect to obtain all possible multi-knockout experiments, a question arises: How many experiments will be needed to successfully identify the CFN which accurately describes the functioning of the system? Utilizing FINE, we address this question and study how its accuracy depends on the complexity of the system in hand.

METHODS

Algorithm Background: the FIN Approach

Experimental data obtained in multi-perturbation studies can give rise to two different kinds of knowledge: (i) *Predictive knowledge*—where given a new, unseen state of the system in hand (a new multi-knockout configuration) one can predict its functioning level, and (ii) *Descriptive knowledge*—where one attempts to reconstruct the functional backbone of the system, *i.e*. describe how the system's components actually interact to perform the function in question. It is the latter kind of knowledge which is the goal of FIN. To this end, it is composed of two parts: (i) Constructing a functional model, which describes how the elements in the system (genes) interact to determine the studied phenotypic behavior, and (ii) Simplifying the resulting functional model (which tends to be very large and unintelligible) and producing a compact, yet accurate, functional description of the system in hand, the CFN.

Constructing a Functional Model from Multi-Perturbation Data Let the investigated system be defined by a pair (N, F). $N = \{1, \ldots, n\}$ is the set of all elements in the system, where each element can be in one of two states, either intact(1) or perturbed(0). $F : \{0, 1\}^n \rightarrow R$, the performance function, associates to every set $S \subseteq N$ a number describing the performance level of the system when the set of elements S is intact, $S = \{x \in N \mid \text{state}(x) = 1\}$. For example, in genetic multi-knockout experiments, N denotes the set of all genes, and for each $S \subseteq N$, $F(S)$ denotes the quantitative phenotype measured in the knockout experiment in which all the genes in S are intact and the rest are knocked-out. A fundamental result from Game Theory shows that $F(S)$ can be uniquely decomposed into the sum $\sum_{T \subseteq S} a(T)$ (Grabisch *et al*., 2000), where the coefficients $a(T)$, denoted *dividends*, describe the marginal contribution of each subset T of the set of intact elements S to the studied performance function F. The dividends are calculated based on the performance levels measured in the different multi-knockout experiments, according to

$$a(S) = \sum_{T \subseteq S} (-1)^{|T|-|S|} F(T), \quad \forall S \subseteq N, \tag{1}$$

(where $|S|$ and $|T|$ denote the cardinality of the sets S and T respectively).

In the context of a data set of multi-knockout experiments with their associated measures performance levels, the dividend computation begins from the dividend of the null group, $a(\emptyset) = F(\emptyset)$ (the performance measured when all the elements are knocked out), and each iteration of Eq.(1) computes the dividend (marginal contribution) of the subsequent supersets. That is, in the second iteration the performance of the single elements minus the performance of the null group is computed, resulting in the marginal contribution of each single element. The third iteration computes the performance of the elements-pairs minus the performance of single elements plus the null group performance, resulting in the marginal contribution of each of the elements-pair, and so forth. Based on the dividends, the performance function F can be represented as a multi-linear polynomial:

$$F(\vec{x}) = \sum_{S \subseteq N} a(S) \cdot \prod_{i \in S} x_i \tag{2}$$

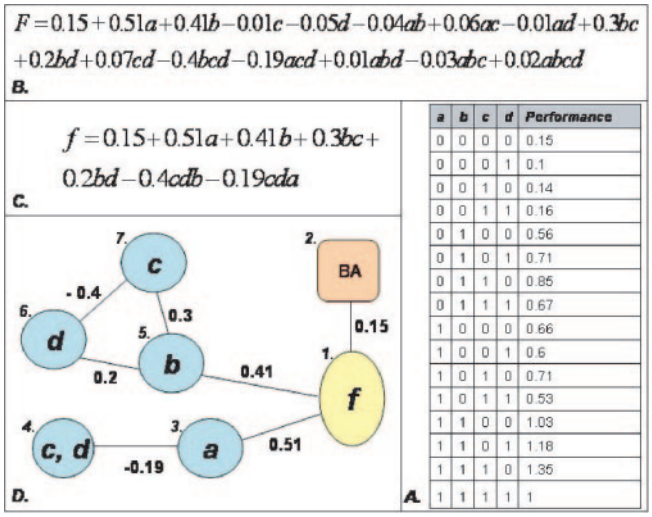
where the vector $\vec{x} \in \{0, 1\}^n$ describes the (intact/knocked-out) states of the elements in the system. Each term in the polynomial, denoted as *summand*, describes a distinct functional pathway since its elements must all be intact to influence the value of F. Obviously if the function is elementary, that is, if there are no dependencies between the elements, it could be fully approximated by a summation over the individual contributing elements (based on n single-knockout experiments). However, in the context of biological systems, such a description is likely to be insufficient and even misleading since such systems are usually complex and involve higher-order interactions.

Constructing the Compact Functional Network (CFN) In the practical analysis of genetic biological data, the full functional description of Eq.(2), is typically very large and unintelligible, containing many 'uninteresting' pathways with very small (but non-zero) influence (dividend). To address this problem, Kaufman *et al*. (2005) introduced the concept of the CFN, a compact representation which approximates the full functional description. The CFN is in itself a multi-linear polynomial which preserves only the most important summands of the full representation.

Figure 1 shows a schematic example of a CFN construction; the full set of all 2^n ($n = 4$) possible multi-knockout experiments is given in box *A*. This set yields a unique performance function F, describing the phenotypic behavior of the system (box *B*). The resulting CFN approximating the full functional description is shown in box *C*. With this compact CFN representation, the approximated performance function f can be visualized in a relatively simple graph (box *D*). This graph, referred to as the *functional diagram*, provides both predictive knowledge, acting as an oracle for the system's behavior at any given state, and descriptive knowledge—explicitly describing the functional structure of the system. Each node in the graph corresponds to a set of (possibly only one) elements and is said to be intact if, and only if, all of its corresponding elements are intact. Additional nodes are the basal activity node *BA* which corresponds to the empty set and the *output node f*. Each simple path which ends at the output node defines a functional pathway (a summand in the CFN) whose elements are those listed on the nodes along the path. The dividend of each such functional pathway is the weight on its first edge. For example, the dividend of the functional pathway (*c-d-b-f*) is the weight on the edge between nodes 7 and 6. Given a knockout experiment, the expected performance level of the CFN can be calculated by summing up the dividends of all the intact functional pathways, that is, sum up the weights on the edges between intact nodes which form a connected component with the output node (for an illustrative example, see legend of figure 1). Note that the existence of an edge between two nodes in the functional diagram does not necessarily imply that they are connected by any physical interaction. Instead, it denotes that there exists a summand in the CFN which contains both these elements, that is, they both participate in a joint functional pathway.

Evidently, the construction of the CFN requires the performance values over all possible multi-knockout experiments; producing such data is an unrealistic demand in most cases. In order to construct a CFN given partial multi-knockout data, the FIN algorithm (Kaufman *et al*., 2005) predicts the performance levels of the missing knockout experiments (using any desired prediction method) and computes the functional model (Eq.(2)) based on these predicted values. It then applies a pruning procedure to remove summands from the functional model, aiming to sustain only the most important ones, while maintaining a pre-defined level of accuracy (comparing the pruned model to the original functional model)[1]. The pruning process is

[1]Throughout the paper, when measuring the accuracy between two continuous vectors p and q, we report the percentage of the variance of p explained by q, this is, $100(1 - (\|p - q\|^2)/(\|p - \bar{p}\|^2))$ where $\bar{p}$ is the mean of p.



a	b	c	d	Performance
0	0	0	0	0.15
0	0	0	1	0.1
0	0	1	0	0.14
0	0	1	1	0.16
0	1	0	0	0.56
0	1	0	1	0.71
0	1	1	0	0.85
0	1	1	1	0.67
1	0	0	0	0.66
1	0	0	1	0.6
1	0	1	0	0.71
1	0	1	1	0.53
1	1	0	0	1.03
1	1	0	1	1.18
1	1	1	0	1.35
1	1	1	1	1

Fig. 1. *A simple schematic CFN construction of a* 4 *element system.* Box *A* provides the analyzed data set of the 16 (2^4) possible combinations of multi-knockout experiments and their corresponding performance measures. Box *B* shows the performance function *F* derived by a dividend analysis of the multi-knockout data set. For example, the dividend value of the subset $\{a, b\}$ is calculated as: Performance($\{a,b\}$) − Performance($\{a\}$) − Performance($\{b\}$) + Performance($\emptyset$) = 1.03 − 0.66 − 0.56 + 0.15 = − 0.04. Box *C* presents the resulting CFN. Finally, box *D* depicts its network visualization. Each of the round-shaped nodes (numbered 3 − 7) corresponds to a set of elements. Node 2 corresponds to the empty set, describing the system's basal activity. Node 1 is the output node. Given a knockout experiment where (for example) only *a*, *c* and *d* are intact then the intact functional pathways are (*a*-*f*) and ([*c*, *d*]-*a*-*f*), and the value of *f* is 0.51 − 0.19 + 0.15 = 0.47, the sum of dividends of intact pathways and the basal activity.

composed of selecting statistically significant summands, and then eliminating those remaining summands which have a low dividend magnitude. The output is the pruned polynomial, composed of the remaining summands.

The FINE Algorithm

Motivation While designed to construct a CFN as accurately as possible, the FIN algorithm still quite frequently produced lengthy and cumbersome descriptions when supplied with biological experimental data. The FINE algorithm was developed to overcome these pitfalls of the FIN for sparse biological systems. Such systems are characterized by an actual small number of important functional pathways in relation to the set of all pathways made possible by different groupings of their elements. Hence, the end goal CFN describing the working of these systems should be small as well. Based on the FIN as a building block, and taking the assumption that the system in hand is sparse, FINE performs an iterative process of pruning and re-approximation of the functional model and produces increasingly more compact and accurate CFN models.

Formal Description of the FINE Algorithm Given a system of n elements, the input of FINE is a set of q multi-knockout experiments with their corresponding experimental performance values (usually, $q \ll 2^n$). FINE is composed of a preprocessing phase of prediction, followed by an iterative process of CFN construction. It identifies the set of important summands and outputs an accurate CFN.

Preprocessing—Constructing the Full Data Predictor. We train a predictor on the accessible, incomplete data set to predict the performance levels of the missing experiments and compute the ensuing dividends of all the 2^n summands (Eq.(2)). This process is done using bootstrapping by randomly resampling with replacement from the available data. Any desired predictor can be used in this process, producing a pair (*D*, *PER*) as output. *D* is a $2^n \times B$ matrix, where $D_{i,j}$ is the estimated dividend of the i^{th} summand according to the j^{th} bootstrap repetition (*B* is the number of bootstrap repetitions), and $PER \in R^{2^n}$, is the predicted performance levels in all possible knockout experiments (taken as the mean prediction over all *B* bootstrap repetitions). We refer to the accuracy of this preprocessing prediction as the *prediction accuracy*.

Iterative Construction of the CFN. This is an iterative process in which the set of summands included in the CFN (i.e. with a non zero coefficient\dividend) is gradually pruned and narrowed down. The input and output of each iteration are pairs of the form (*D*, *PER*), as defined above. On each iteration, the number of non-zero rows in *D* is monotonically reduced.
Each iteration step is comprised of two phases:

(1) *Summands selection phase*—For each summand, we use the corresponding row in *D* to calculate two indices: (i) The significance level of its dividend (based on a *t*-test where the null hypothesis is that the dividend magnitude is zero), and (ii) The expected magnitude of its dividend (taking the mean value). The most important summands are then chosen based on these indices, using forward selection and backward elimination procedures. These procedures are controlled by two pre-determined target levels of accuracy, $level_1 > level_2$[2]. Starting from an empty summand set (an empty CFN), we gradually add summands to the CFN, doing so by the order of their dividends' significance, until we reach a desired accuracy of $level_1$. Next, we apply backwards elimination on the resulting set of significant CFN summands, now eliminating summands by the order of their dividends' magnitude (starting from the small ones) until the lower limit of accuracy, $level_2$, is reached.

(2) *Dividends recomputation phase*—Let *m* denote the cardinality of the set of important summands as of the preceding summands selection phase. We fit each of the *m* chosen summands a new dividend coefficient according to the following model: $Q \cdot \vec{\delta} = \vec{y}$, where *Q* is a binary $q \times m$ matrix describing the partial set of biological knockout experiments in hand, defined as: $Q_{j,i} = 1$ iff $T_i \subseteq S_j$, where T_i is the set of elements included in the i^{th} important summand, and S_j is the subset of genes intact in the j^{th} experiment. $\vec{y}$ is the given $q \times 1$ vector of observed performance levels. The coefficients vector $\vec{\delta}$ is therefore the new estimated dividends vector. Clearly, the number of free variables in this model decreases in each iteration since the set of important summands is monotonically reduced. When the matrix *Q* does not have a full rank, there is obviously no unique solution. The particular basic solution chosen is determined using the QR factorization with column pivoting (Businger *et al*., 1965). An over determined equation set is typically reached after a small number of iteration steps (on our simulations, the majority of cases did not require more than 5 iteration steps to reach an over determined equation set). Repeating the calculation of $\vec{\delta}$ using bootstrapping results in a new set of dividends, *D*, from which *PER* is calculated and both serve as the input to the next iteration.

The iterative process continues until the following stopping criteria is satisfied: either the given model cannot be pruned (*i.e.,* the output of an iteration is equal to its input) or that a user defined upper bound on the number of iterations is reached. (the upper bound of 10 iterations, used in our simulations, was reached in approximately 1% of the experiments).

The algorithm returns the output of the last prediction phase: a predicted set of all 2^n knockout experiments (*PER*) and a multi-linear polynomial whose coefficients (most of them zero) are taken as the mean values over the rows of *D*. This is a CFN representation of the given performance

[2]The accuracy level is computed between the prediction based on the chosen dividends and the prediction given by the previous iteration.

function in which all remaining summands (those with non zero dividend coefficients) are important per our definition and whom cannot be further eliminated.

To visualize the output of FINE, we have developed an automated module for the construction of functional diagrams, based on a series of factorization steps applied on the CFN polynomial. Due to space limitations details are not provided, however, this module is available as a part of the FINE software package.

RESULTS

Measures for Evaluating FINE

We present a set of measures, testing to what extent does FINE achieve its objectives. These measures evaluate the accuracy of the CFN obtained using partial knockout data to that obtained with full data. We define: *Ground truth performance*—the vector of all 2^n knockout experiments' performance levels, as given by the predicted performance function. *Ground truth CFN (GTCFN)*—the CFN obtained by applying FINE to the full data set of *ground truth performance* values. The *CFN performance* is a vector of all 2^n performance levels computed from a given CFN over all possible knockout experiments. Based on these definitions we present the following measures to quantify CFN accuracy:

- **Operational accuracy**—the ability to produce accurate predictions of the system's behavior at any given state, measured by the match between the ground truth performance and the performance values predicted by the CFN.
- **Dividend accuracy**—the accuracy of the weights assigned to each functional pathway, measured by the match between all 2^n dividends (some are zero) of the CFN and the GTCFN.
- **Descriptive accuracy**—the ability to detect the most important pathways. Since the GTCFN, by construction, contains only the most important summands of the original target function (such that the pre-defined level of accuracy is satisfied) it can be used as a ''gold standard'' for measuring the descriptive accuracy. We therefore compare the CFN summands to the GTCFN summands through the following measures:
- *Specificity*—the total magnitude of CFN dividends whose corresponding summands appear in the GTCFN, divided by the total magnitude of all CFN dividends.
- *Sensitivity*—the total magnitude of GTCFN dividends whose corresponding summands are included in the CFN, divided by the total magnitude of all the GTCFN dividends.
- *Jaccard coefficient*—the number of CFN summands which appear in the GTCFN (tp), divided by the combined number of GTCFN summands (t) and the CFN summands which do not appear in the GTCFN (fp). This score reflects the 'conjunction over union' between the CFN summands and GTCFN summands, $(tp/(t+fp))$.
- *Top summands detection rate*—the success rate in identifying the three most important summands in a given performance function, where each summand is ranked proportionally to the number of multi-knockout experiments on which it affects. In a system of n elements, the rank of a summand with s elements and a dividend value of d is set to $|d| \cdot 2^{n-s}$.

Combined, these measures provide a comprehensive evaluation of the performance of FINE. Overall, FINE should give an accurate approximation of the actual function investigated, (*operational accuracy*), in which the important subsets of elements are expressed (*descriptive accuracy*) with the accurate weights assigned to them (*dividend accuracy*). Note that the *operational accuracy* is different from the *prediction accuracy* as the former relates to the preliminary prediction and the latter, to the output of FINE.

FINE Analysis of Simulated Data: Descriptive Vs. Predictive Accuracy

First, a comparison of FINE with FIN in the analysis of sparse systems is in hand. Figure 2 illustrates the continuous improvement in both descriptive accuracy and dividends accuracy throughout the iterative process of FINE, measured in our simulation experiments. Evidently, the more sparse the system is, the more significant is the improvement along the iterative process. These results clearly demonstrate the superiority of FINE over the FIN algorithm in sparse systems (as the FIN is equivalent to FINE with a single iteration).

Our main focus is to utilize FINE to attend the following fundamental questions: having obtained multi-knockout performance data of some cellular function, how well can we expect to understand and describe it's processing? In terms of this paper, how is the descriptive accuracy of a CFN produced by FINE dependent on the prediction accuracy of the data that has been collected? Furthermore, how and to what extent is this relation dependent on the architecture of the underlying network, *i.e.* the studied performance function? In biological systems, these underlying networks are currently mostly unknown. Therefore, the dependence of the operational and descriptive accuracy on the prediction accuracy is important, since prediction accuracy is the only measure one may have in hand. To study these questions in depth, we perform a comprehensive set of experiments using simulated multi-knockout performance data.

The Simulation Experiments We generate a set of random performance functions, each inducing a different functional backbone network architecture. These functions are multi-linear polynomials[3], parameterized by (n, m, c): having n elements, with m summands (functional pathways), each summand containing no more than c elements (the length of the pathways is bounded by c). We study a wide range of functions, varying form simple ($n = 8$, $m = 2$, $c = 2$) to more complex ($n = 8$, $m = 16$, $c = 8$). For each parameter set (n, m, c) we consider 10 random performance functions, each inducing a different network architecture. The coefficients of each polynomial are selected randomly from a uniform distribution (on the interval [6, 10]) and arbitrarily assigned with a ± sign. The input data to FINE is a set of 'knockout experiments' obtained by considering different intact subsets out of the n elements and calculating their corresponding performance levels. For each of the random performance functions, we performed a set of FINE analyses using a span of partial input data sets, ranging from 10 to 256 samples (out of $2^8 = 256$) yielding a wide range of prediction accuracies. In the current implementation we used k-nearest neighbors (KNN) as the 'default' prediction method (used in the preprocessing stage) with the parameter k set to 3. The target levels of accuracy, $level_1$ and $level_2$, were set to 98% and 95% respectively.

[3]Our choice of multi-linear polynomials as target models stemmed from the fact that performance levels of any multi-knockout data set can be uniquely described in such canonical form.

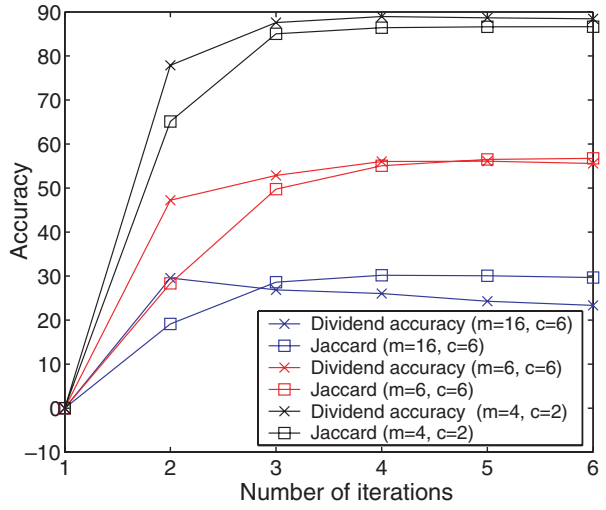


Fig. 2. *Convergence and improvement in accuracy across FINE selection and recomputation iterations.* The figure depicts the improvement of the Jaccard coefficient and the dividends accuracy (y-axis) across the algorithm's iterations (x-axis) until convergence. Data presented for a set of performance functions with three different parameter sets (detailed on the inset of the figure), where each parameter set defines a different level of sparseness (see next section). The continuous improvement in the Jaccard coefficient and dividend accuracy implies that we continuously eliminate more false positive (*fp*) summands than true positive (*tp*) ones (Jaccard coefficient = $tp/(t+fp)$) and that the dividend coefficients assigned to these summands are increasingly more accurate. The point of reference (on x = 1) is the result of the first iteration, or equivalently, the result of the FIN algorithm.

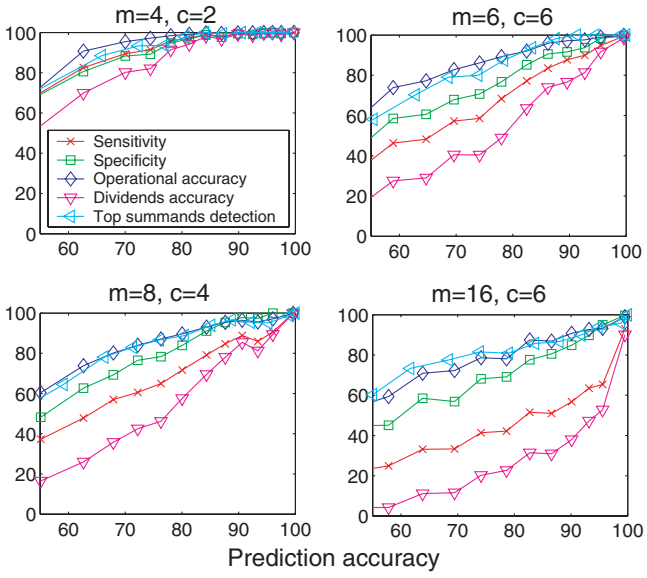


Fig. 3. *Performance of FINE on simulated data.* The different performance measures (y-axis) are plotted against the prediction accuracy of the input given to the algorithm (x-axis). Four different performance functions with different parameter sets are displayed.

The Simulation Results Figure 3 demonstrates the simulation results. The different performance measures are plotted against the prediction accuracy of the sample sets. Evidently, all the performance measures increase with the prediction accuracy. Notably, in the more complex functions, the descriptive accuracy measures rise to high levels only at fairly high prediction accuracy levels. This implies that when the assumed size and complexity of the biological system studied is considerable, one must seek to gather ample data ensuring high levels of prediction accuracy, otherwise an accurate descriptive identification of the system is unlikely (at least with FINE). Interestingly, in all the performance functions tested, regardless of their complexity, the operational accuracy of FINE is higher than the prediction accuracy of the initial prediction method. This fact implies that FINE, in addition to reconstructing the functional backbone, also acts as a smart predictor, which utilizes the assumption that the system in hand is functionally sparse to yield improved predictions of the behavior of the system in unknown states. Another interesting perspective on FINE's performance is given by the top summands detection rate measure; evidently, when the prediction accuracy rises above 75%, the top summands detection rate is higher than 80% even in the more complex cases ($m = 16$, $c = 6$). Compared with the performance of the FIN algorithm throughout our simulation experiments, FINE achieves better results in 96.4% of the cases, both in terms of descriptive accuracy (measured by the Jaccard coefficient) and dividends accuracy.

Figure 4 presents the prediction accuracy required for obtaining a desired level of descriptive accuracy, as a function of the complexity of the performance function. The results show a clear monotonic dependency of the required prediction accuracy on both the number and length of summands of the target functions. In biological applications, once a set of experiments has been performed, the prediction accuracy can be evaluated. Thereafter, by assuming the number and lengths of pathways taking place in the function studied, an estimate of the achievable CFN descriptive accuracy can be obtained. However, some caution is warranted since different predictors yield CFNs with different descriptive accuracies subject to the same initial prediction accuracy. Yet, the relative ordering between predictors is conserved across a span of prediction accuracy levels (data not shown).

FINE Analysis of the *Endo16 Cis*-regulatory System

To study the workings of FINE with biological data, we focus on the computational logic model constructed for the *cis*-regulatory system of the *endo16* gene of the sea urchin, *Strongylocentrotus purpuratus* presented by Yuh *et al.* (2001). This *cis*-regulatory system was studied thoroughly in a series of studies (*e.g.* Yuh *et al.*, (1996), 1998, 2001)). Combining the knowledge assembled by these studies allowed the formulation of a computational model (Yuh *et al.*, 2001) which describes in detail how the activity of the *endo16* gene is determined by it's *cis*-regulatory elements (transcription factor (TF) binding sites).

The main elements of the *endo16 cis*-regulatory system can be divided into three distinct groups. The two main groups, referred to as module A and B, correspond to two sets of TF binding sites, lying on two adjacent regions of the *cis*-regulatory apparatus. The elements in the third group correspond to whole clusters of binding sites (modules) lying upstream of modules A and B. Yuh *et al.* (2001) show how the elements in these groups interact to determine the expression level of the *endo16* gene throughout embryogenesis. Early in development, the *endo16* gene participates in the speci-

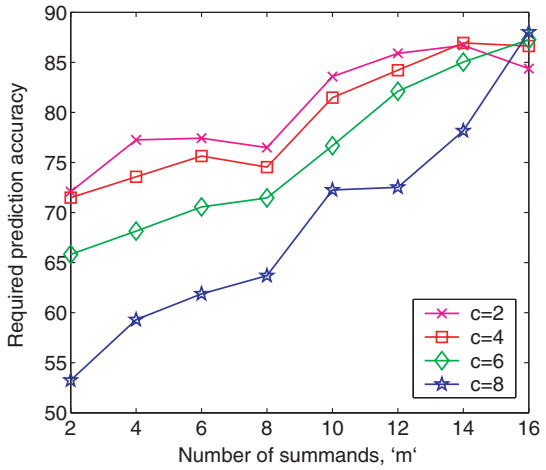


Fig. 4. *Performance of FINE on an ensemble of performance functions.* For each performance function studied we note the input prediction accuracy needed to achieve a desired descriptive accuracy—a combination of specificity above 55% and sensitivity above 75% (the y-axis). The figure depicts the dependency of the descriptive accuracy as a function of prediction accuracy over a range of different values of m, that is, over different number of summands in the performance function. Each line corresponds to a different value of c, that is, to a different maximum permitted length of summands in the performance function. For example, for a performance function with the parameters of ($m = 6$, $c = 4$), the prediction accuracy needed to achieve the descriptive accuracy criteria stated above is 70%. Observe, that the required prediction accuracy shows a monotonic increase both with c and m.

fication events that define the endomesoderm. This function is mainly dependent on module A. Later on, it serves as a gut-specific differentiation gene. This function is mainly dependent on module B. However, it still requires module A, whose main role at that point is to act as a mediator between module B and the basal transcription apparatus. In parallel, the upstream modules (referred to as modules F, E, D and C) mediated by a certain binding site in module A, were shown to serve as a repression subsystem whose role is to force spatial constraints over the activity of the *endo16*.

The computational model of Yuh *et al*. explains this spatial and temporal dependent activity as a direct result of the intactness of the elements in the three groups and of the concentration levels of the TFs which bind in them. Specifically, there are three TFs considered as *kinetic drivers*, in the sense that their activity profiles provide continuous, time-varying input to the system, whereas the rest of the TFs are perceived in only two discrete states—active or inactive.

In this study we aim to characterize the *endo16* regulation via FINE. We focus on a single, central, time point of 60 hours after fertilization, which is after the switch between module A and B took place (Yuh *et al*., 2005). The perturbation data is obtained by considering the effect of mutating or functionally knocking out different sets of binding sites. Our first task is to construct the full functional model and the Ground Truth CFN of the *endo16* system and use the resulting functional diagram to draw conclusions about the functional structure of the system. We then show how our observations match those given by Yuh *et al*. (2001) Our second task is to show how well can FINE reconstruct the GTCFN with limited training data. Finally, we compare the performance of FINE to that of the FIN algorithm.

We consider nine input variables, eight of them represent single binding sites—Otx, P and CG1 sites of module A, CY, CB1, CB2, UI and R sites of module B and a single variable Φ denoting the spatial repression subsystem, composed of the Z site of module A and the upstream modules F, E, D and C (Yuh *et al*., 1996, 1998). Each binding site variable can be assigned either with a value of '1' indicating that it is present and is occupied by its respective TF, or with a value of '0', indicating that it has been mutated or that its respective TF was inactivated or eliminated. The variable Φ is assigned with a value of '1' if and only if the repression system is inactive.

In order to accurately compute the dividend decomposition of the system, we need to obtain the activity levels of the *endo16* in response to all 2^9 perturbation configurations (Eq.(1)). We estimate the values of the basal promoter activity and of the concentration levels of the kinetic drivers (which bind at the UI, CB2 and Otx sites) at the time point of 60 hours after fertilization, reflecting their relative magnitudes in accordance with Yuh *et al*. (2005) (UI = 1, Otx = 0.2, CB2 = 0.3, Basal Activity = 0.2).

These values were then used to query the computational model for the corresponding activity levels. This provides the perturbation data needed for obtaining the full FIN functional model, via Eq.(2). In terms of our parameterization, the parameters of this functional model are ($n = 9$, $m = 7$, $c = 7$). We then apply FINE to obtain the GTCFN—a compact representation of the functional backbone of the *endo16 cis*-regulatory system. Two out of the seven summands were pruned during the GTCFN construction.

Figure 5 presents the functional diagram, obtained from the full functional model. Nodes 3 to 9 corresponds to different subsets of *cis*-regulatory elements. Node 2 correspond to the basal activity and node 1 is the output node. Each weighted edge corresponds to a dividend value. Dashed edges have zero weight and serve as Boolean *and* operators. The diagram shows a clear distinction between module *A* (nodes 3-4, squares), *B* (nodes 6-8, hexagons) and the hybrid subsystems (nodes 5, 9 ovals). The functional diagram of the GTCFN is, naturally, a subgraph of the full model's diagram, and can be obtained by excluding nodes 7 and 9 (indicated by a gray filling).

The functional diagram, based on the automated visualization module, clearly outlines the functional structure of the system and allows us to draw various insights regarding the different logical subsystems (or functional pathways) involved in determining the expression of the *endo16*. We point out a few such observations: (i) Node 5 acts as a bottle neck for the output of nodes 6-9. It depends on the sites P, CG1 and CB2. If one of these three elements is assigned with a value of zero, then the output of the system will depend solely on node 4 (the Otx site of module A). This subsystem is recognized by Yuh *et al*. as the *linkage subsystem*, which connects the output of module B into module A. (ii) The Otx site participates in two counteracting functional pathways, starting at nodes 9 and 4; If the pathway including nodes (9-6-5-3-1) is intact then the Otx site has no influence on the system, since its positive contribution via pathway (4-3-1) is totally repressed. This subsystem is recognized by Yuh *et al*. as the *BA intermodule input switch*, which represses the output of module A (via the Otx site) and leaves it solely as a mediator for the output of module B. (iii) The input elements in node 8 play only a single role in the system, which is to increase the output of node 6 by two fold. Yuh *et al*. term this as the *synergism*

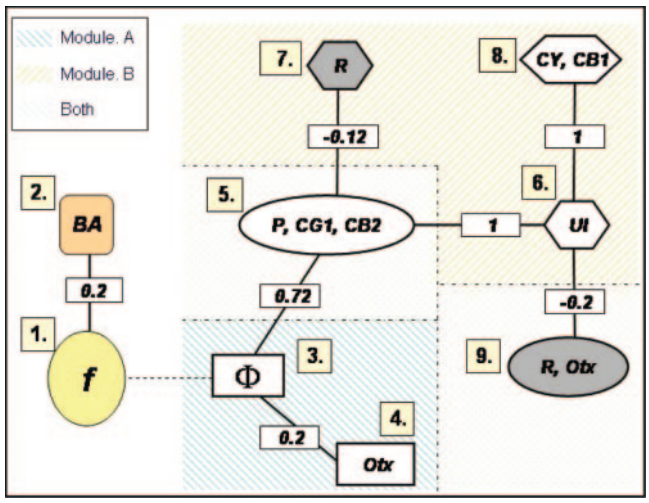


Fig. 5. *Functional diagram of the computational model of the* endo16 *cis-regulatory system.*

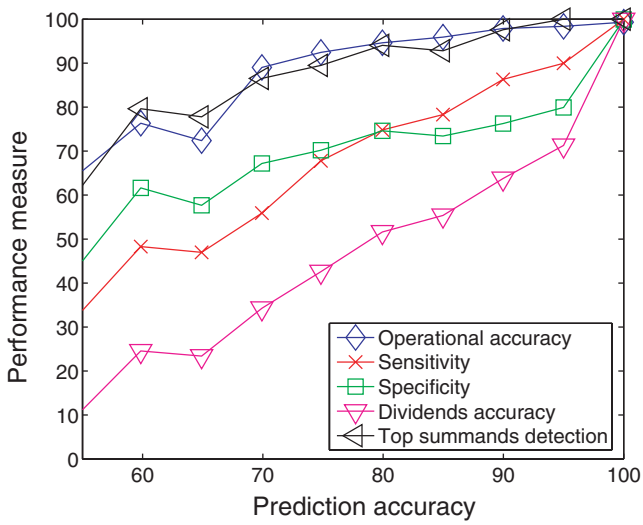


Fig. 6. *Performance of FINE analysis of the computational logic model of the* endo16 *cis-regulatory system.* The different FINE performance measures are plotted against the prediction accuracy of the data samples.

subsystem, which, mediated by the CY and CB1, steps up the output of UI. (iv) Taking a wider perspective, we see that module A is directly connected to the output node (assuming that the repression subsystem is inactive) whereas module B requires the intactness of module A in order to have an effect. On the other hand, the quantitative influence of module A alone (via node 4) is of low magnitude, compared to that of module B. This is in agreement with the fact that the role of module A at the time point we selected is mainly to serve as a mediator for the output of module B, whereas its own output is of less importance. Evidently, each of these observations, based on the functional diagram has an equivalence in the logical analysis of Yuh *et al*. This fact illustrates the utility of the FIN approach (and the FINE algorithm) as a tool for representation and analysis of biological systems.

The two summands which were pruned during the GTCFN construction correspond to the functional pathways connecting nodes 7 and 9 to the output node. The biological phenomena which correspond to these summands are both related to the R site: (i) The slight increase in the CB2 output which occurs once the R site is mutated (node 7). (ii) The BA intermodule input switch (node 9). Both these subsystems were recognized to have a marginal influence on the expression of the *endo16* at the time point examined (Yuh *et al*., 2001), and indeed, removing the two summands from the functional model reduces the operational accuracy by a mere 3.74%.

To study the relation between prediction accuracy and descriptive accuracy in the *endo16* system, we apply an assay similar to the one described in the simulated data section: We apply the FINE algorithm to a set of random samples of different sizes drawn out of the set of all 2^9 perturbation configurations. Figure 6 displays the performance of FINE as a function of the prediction accuracy yielded by the various data sets, in a manner analogous to that of Figure 3. Evidently, the results are quantitatively similar to the results on the simulated data (testifying that the simulated networks behave in a similar manner, CFN-wise, to the biological model).

The three most important summands in the GTCFN, according to the ranking scheme defined for the top summands detection rate measure, correspond to the pathways connecting the output node to nodes 2, 5 and 6 (pathways ($BA - f$), ($[P, CG1, CB2] - \phi - f$), ($UI - [P, CG1, CB2] - \phi - f$)). Indeed, the corresponding subgraph (induced by nodes 1, 2, 3, 5 and 6) includes the most influencing subsystems—the UI and CB2 sites where the two dominant kinetic drivers at the selected time point bind at, the *linkage subsystem*, which connects the output of module B into module A, and the basal activity which naturally effects the expression of the gene in all possible perturbation configurations and is therefore considered important as well. Interestingly, FINE identifies this subgraph, even when the prediction accuracy is poor. For example, with prediction accuracy of 60%, the top summands detection rate is 80%.

Comparing the performance of FINE to that of the FIN algorithm shows a clear superiority of the former. FINE achieves higher rates of both descriptive accuracy and dividends accuracy in 94.3% of our simulation experiments, producing significantly more compact and accurate models (data not shown).

CONCLUSIONS

This paper addresses a new challenge within the context of gene network analysis. In contrast to prevalent approaches, which aim to reveal the network of interactions between genes, our goal is to identify the underlying functional network. In this network description, the genes' states determine a quantitative phenotype of the network, and its architecture visualizes and explains how the studied function is actually carried out. To this end, we rigorously study the capabilities of FINE, a new, iterative algorithm based on the FIN algorithm presented in Kaufman *et al*. (2005). It is designed to handle sparse networks and leverages this assumption to achieve improved results. It is shown to successfully analyze simulated complex networks utilizing multi-knockout data, obtaining a simple and compact description of the underlying functional network. This compact representation delineates the important gene sets (pathways) and their functional influence. Our results demonstrate that in small-scale systems (of the scale of multi-knockouts currently

studied in biology), FINE successfully identifies the main subsets of genes with a low rate of false positives, obtaining a high success rate in identifying the top 3 important pathways in a system. Similar results are achieved in the biological example of the *endo16* regulation where FINE successfully extracts *in an automatic manner* the main known biological modules of the *cis*-regulatory network. The success rate in identifying the top three main functional modules in this system is above 80%, even with a predictive accuracy of 60%. Evidently, FINE outperforms the FIN when applied to sparse systems; this occurred in 96.4% of the simulated experiments we have conducted. However, in cases where the network architecture transpires such that two pathways completely cancel out the functional effect of each other in an almost precise manner, FIN might outperform FINE (such cases occurred in approximately 3% of our simulations). In addition, if the underlying system is not sparse and is composed of many functionally interacting pathways, FINE may lead to an erroneous, grossly over-pruned network, while FIN is likely to lead to a significantly more accurate description.

Applying the FINE visualization module in the analysis of the biological example, demonstrates the correspondence between insights that can be gained via the functional diagram and the pertaining biological knowledge of the *endo16* regulation, summarized in its Boolean logic description.

The second main theme addressed in this study is the question of how many experiments are needed to successfully identify the CFN. Our results show that the descriptive and operational accuracy of the CFN are dependent on the prediction accuracy of the experimental set in hand. However, the complexity of the underlying network further modulates the required prediction accuracy in a significant manner. On a more quantitative level, the simulated data results provide the biologist with general ballpark numbers as to what levels of prediction accuracy he must achieve to obtain a desired level of descriptive accuracy. To this end, however, the biologist must have some a-priori gross estimation of the complexity/architecture of the system in hand.

FINE is not limited to small-scale systems, and is potentially scalable to much larger networks, under some constraints; We are now developing and studying a *k-bounded* variant of the FINE algorithm, in which we assume that, even if the system is large, only a bounded set of k elements significantly influences the investigated function (and different tasks in the system may be realized by different, possibly overlapping bounded sets). Other directions for future work include applying the FINE to multiple functions concomitantly—in such cases one can identify and classify functions according to their functional backbones or extract common features within the obtained functional backbone. Importantly, if the multiple functions under investigation are assumed to share similar functional modules the construction of the CFN can benefit from a higher degree of accuracy by validating the importance of the chosen dividends across the different functions.

Since FINE is model independent, it is potentially applicable to a wide variety of systems. The only requirement is that the function performed by the system can be measured under different discrete states of its elements e.g. perturbed, silenced, inactive, enhanced, over-expressed and so on. Notably, the 'elements' perturbed need not necessarily be single elements and can be sub-modules of a system. For example, in the sea urchin model, we can relate to the *endo16* CFN as a single node in a more comprehensive developmental network, leading to a hierarchical functional view of the system. It is likely that the most immediate and rewarding current application of FINE is in the analysis of multi-perturbation studies in genetics, in view of the rapid recent advances in gene silencing with RNAi. The FINE algorithm offers a viable way for the accurate identification of the main functional pathways in biological systems.

ACKNOWLEDGEMENTS

N.Y. is supported by the Tel-Aviv university president and rector scholarship, A.K. is supported by the Yeshaya Horowitz Association through the Center of Complexity Science. E.R's research is supported by grants from the Israeli Science Foundation (ISF), the Yeshaya Horowitz Center of Complexity Science, and from the Tauber Fund.

REFERENCES

Carpenter,A.E. and Sabatini,D.M. (2004) Systematic genome-wide screens of gene function. *Nat. Rev. Genet.*, **5**, 11–22.

Barabasi,A.L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, **5**, 101–113.

Davidson,E.H. *et al.* (2002) A genomic regulatory network for development. *Science*, **295**, 1669–1678.

Gu,Z. *et al.* (2003) Role of duplicate genes in genetic robustness against null mutations. *Nature*, **421**, 63–66.

Kaufman,A. *et al.* (2005) Quantitative analysis of genetic and neural multi-perturbation experiments. *PLoS Comput. Biol.*, **1**(6): e64.

Tong,A.H. (2004) Global mapping of the yeast genetic interaction network. *Science*, **303**, 808–813.

Yuh,C.H., Bolouri,H. and Davidson,E.H. (2001) cis-regulatory logic in the endo16 gene: switching from a specification to a differentiation mode of control. *Development*, **128**, 617–629.

Kaufman,A., Kupiec,M. and Ruppin,E. (2004) Multi–knockout genetic network analysis: The Rad6 example. *Proceedings of IEEE Computational Systems Bioinformatics Conference (CSB'04)*, pp. 332–340.

Hammond,S.M., Caudy,A.A. and Hannon,G.J. (2001) Post-transcriptional gene silencing by double-stranded RNA. *Nature Rev. Gen.*, **2**, 110–119.

Ideker,T.E., Thorsson,V. and Karp,R.M. (2000) Discovery of regulatory interactions through perturbation: inference and experimental design. In *Proc. of the Pac Symp. on Biocomputing*, pp. 305–316.

Pe'er,D. *et al.* (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics* 17 Suppl 1:S215–S224.

Ideker,T.E. *et al.* (2001) Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, **292**, 929–934.

Tegner,J. *et al.* (2003) Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proc. Natl. Acad. Sci. USA*, **100**, 5944–5949.

Keinan,A. *et al.* (2004) Fair attribution of functional contribution in artificial and biological networks. *Neural Computation*, **16**, 1887–1915.

Thieffry,D. *et al.* (1998) From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in escherichia coli. *BioEssays*, **20**, 433–440.

Jeong,H. *et al.* (2000) The large-scale organization of metabolic networks. *Nature*, **407**, 651–654.

Grabisch,M., Marichal,J.L. and Roubens,M. (2000) Equivalent representations of a set function with applications to game theory and multicriteria decision making. *Mathematics of Operations Research*, **25**, 157–178.

Businger,P.A. and Golub,G.H. (1965) Linear least squares solution by householder transformation. *Numer. Math.*, **7**, 269–276.

Yuh,C.H. and Davidson,E.H. (1996) Modular *cis*-regulatory organization of endo16, a gut-specific gene of the sea urchin embryo. *Development*, **122**, 1069–1082.

Yuh,C.H., Moore,J.G. and Davidson,E.H. (1996) Quantitative functional interrelations within the *cis*-regulatory system of the s. purpuratus endo16 gene. *Development*, **122**, 4045–4056.

Yuh,C.H., Dorman,E.R. and Davidson,E.H. (2005) Brn1/2/4, the predicted midgut regulator of the endo16 gene of the sea urchin embryo. *Dev. Biol.*, **281**, 286–298.

Yuh,C.H., Bolouri,H. and Davidson,E.H. (1998) Genomic *cis*-regulatory logic: Functional analysis and computational model of a sea urchin gene control system. *Science*, **279**, 1896–1902.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e547–e556
doi:10.1093/bioinformatics/btl261

Accessing bioscience images from abstract sentences

Hong Yu[1,*] and Minsuk Lee[2]

[1]Department of Health Sciences and [2]CEAS, University of Wisconsin-Milwaukee, Wisconsin

ABSTRACT

Images (e.g., figures) are important experimental results that are typically reported in bioscience full-text articles. Biologists need to access images to validate research facts and to formulate or to test novel research hypotheses. On the other hand, biologists live in an age of information explosion. As thousands of biomedical articles are published every day, systems that help biologists efficiently access images in literature would greatly facilitate biomedical research. We hypothesize that much of image content reported in a full-text article can be summarized by the sentences in the abstract of the article. In our study, more than one hundred biologists had tested this hypothesis and more than 40 biologists had evaluated a novel user-interface BioEx that allows biologists to access images directly from abstract sentences. Our results show that 87.8% biologists were in favor of BioEx over two other baseline user-interfaces. We further developed systems that explored hierarchical clustering algorithms to automatically identify abstract sentences that summarize the images. One of the systems achieves a precision of 100% that corresponds to a recall of 4.6%.

Contact: hong.yu@dbmi.columbia.edu

1 INTRODUCTION

The rapid growth of electronic publications in bioscience has made it necessary to create information systems that allow biologists to navigate and search efficiently among them. *PubMed* is an information retrieval system that returns a list of documents in response to users' queries. *Arrowsmith* helps biologists uncover biologically significant relations between two previously disparate fields of inquiry (Smalheiser and Swanson 1998). *BioText* is an information retrieval system that allows biologists to refine the retrieved MEDLINE articles and to categorize the retrieved articles based on the MeSH terms that were assigned to the articles (Hearst 2003). *GeneWays* is an information extraction and visualization system that extracts from literature molecular interactions related to pathways (Rzhetsky *et al*. 2004). *iHOP* is an online service that identifies sentences that relate two genes (Hoffmann and Valencia 2005). *BioMedQA* is a question answering system that provides short text in response to questions posed by biomedical researchers and physicians (Yu *et al*. 2006). See the review article (Jensen *et al*. 2006) for other information systems. Additionally, there are numerous annotated databases, e.g., SWISSPORT, OMIM (Hamosh *et al*. 2005), and BIND (Alfarano *et al*. 2005), that provide different levels of annotated literature information about genes and molecular interactions.

Most of the information systems, however, target text information only and ignore other important data such as images. Images (e.g., figures) are usually the ''evidence'' of biological experiments. An image is worth a thousand words. Biologists need to access image data to validate research facts and to formulate or to test novel research hypotheses. For example, a biologist may want to see the image (Figure 1) that supports the fact that ''a stem cell can generate sebaceous glands.'' Additionally, full-text articles are frequently long and typically incorporate multiple images. For example, we have found an average of 5.2 images per biological article in the journal Proceedings of the National Academy of Sciences (PNAS). Biologists need to spend significant amount of time to read the full-text articles in order to access specific images.

In order to facilitate biologists' access to images, certain online journal publishers (e.g., Science direct) introduce a service called SummaryPlus (as shown in Figure 2) which lists images and their captions that appear in the full-text article. Such presentation has the promise of improvement over the traditional single-document-per-article format that has dominated bioscience publications since the first scientific article appeared in 1665 (Gross 2002).

We hypothesize that we can further enhance the SummaryPlus user-interface design. For example, the current SummaryPlus user-interface does not show any connections between images; this is contradictory to the fact that images reported in a full-text article are not disjointed. On the contrary, images are related to each other and typically, as a whole, leads to the conclusion of the full-text paper. Additionally, the associated text other than an image caption is frequently useful to illustrate the image content.

By working with hundreds of biologists, we conclude that much of the image data that appear in a full-text article can be summarized by the sentences in the abstract of the full-text article. Because biologists must read the abstract in order to understand a full-text article; linking abstract sentences to images will be the most effectively and convenient way for biologists to access images. This study reports our design and evaluation of BioEx (as shown in Figure 4), a user-interface that links abstract sentences to images. We further explored natural language processing approaches, in particular, hierarchical clustering to automatically link abstract sentences to images.

2 DO ABSTRACT SENTENCES CORRESPOND TO IMAGES?

We hypothesize that images reported in a full-text article can be summarized by sentences in the abstract. To test this hypothesis, we randomly selected a total of 329 biological articles that are recently published in four journals Cell (104), EMBO (72), Journal of Biological Chemistry (92), and Proceedings of the National Academy of Sciences (PNAS) (61). For each article, we emailed the corresponding author and invited him or her to identify abstract sentences that summarize image content in that article. In order to

*To whom correspondence should be addressed.

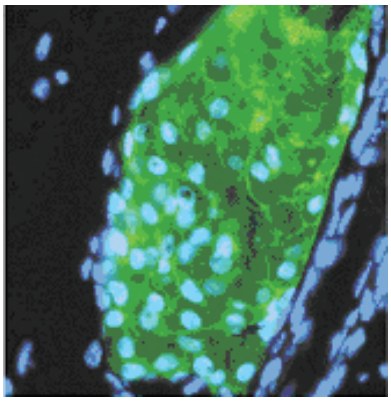


Fig. 1. Sebaceous glands genereted by the transplanted progeny of a single multipotent stem cell isolated from a rat whisker follicle. The picture appears in the cover page of PNAS 102(41): 14477–14936.

Fig. 2. The summaryPlus user interface.

eliminate the errors that may be introduced by sentence boundary ambiguity, we manually split abstract sentences and sent the sentences as the email attachments.

A total of 119 biologists from 19 countries participated voluntarily the annotation to identify abstract sentences that summarize figures or tables in their publications, this resulted in a total of 114 annotated articles (39 Cells, 29 EMBO, 30 Journal of Biological Chemistry, and 16 PNAS), a collection that is 34.7% of the total articles we requested. The responding biologists included the corresponding authors to whom we had sent emails, as well as the first authors of the articles to whom the corresponding authors had forwarded our emails. None of the biologists were compensated.

This collection of 114 full-text articles incorporates 742 figures, 75 tables, and 826 abstract sentences. The average number of figure or table per document is 7.2±1.7 and the average number of sentences per abstract is 7.2±2.0. Our data show that 87.9% figures and 85.3% tables correspond to abstract sentences and 66.5% abstract sentences correspond to images; those statistics have empirically validated our hypothesis that image content can be summarized by abstract sentences. Since an abstract is a summary of a full-text article, our results also empirically validate that images are important content in full-text articles.

Table 1. The numbers of links between abstract sentences to images

Type	Number of Links
1:1	151
1:2	145
1:3	53
1:4	26
1:5	9
1:6	4
1:7	1
2:1	173
3:1	36
4:1	14
5:1	2

1:1: An abstract sentence is linked to only one image and the image is only linked to the abstract sentence.
1:N: An abstract sentence is linked to N images, N > 1.
N:1: N abstract sentences are linked to one image, N > 1.

Note that the total number of tables is a small fraction (10.1%) of the total number of figures. Furthermore, out of the four journals, only EMBO includes Table as images. The total number of table images in our data collection is 15, which represents only 2% of the total image files. We have therefore focus only on the 742 figure images in this study.

We identified three types of links between abstract sentences and images. *One-to-one* is defined as an abstract sentence that is linked to only one image and the image is only linked to the abstract sentence. *One-to-many* is defined as an abstract sentence that is linked to two or more images. *Many-to-one* is defined as an image that is linked to two or more abstract sentences. Table 1 shows the numbers of the three categories in our 114 annotated full-text articles.

After manually examining the annotated articles, we found that we could approximately group full-text articles into four link patterns (examples are shown in Figure 3) based on the positions in which abstract sentences or images orderly appear in the abstract or the full-text articles. In Figure 3A, the abstract sentences are aligned with images in the order they appear in the full-text articles. Figure 3B shows that abstract sentences do not correspond to images in the order they appear in the full-text articles. Figure 3C shows that images are linked to only a few abstract sentences. Figure 3D shows that some images are aligned with images in the order they appear in the full-text articles and some do not. We speculate that the link patterns may be useful as additional features for inference authorship. Previously, word frequency has been explored for this task (Mosteller and Wallace 1963). On the other hand, the irregular alignment has made the task of automatically mapping abstract sentences to images more challenging, which will be discussed in Section 4.

3 BIOEX USER-INTERFACE DESIGNS AND EVALUATION

We have shown in Section 2 that biologists have judged that 87.9% images in the total of 114 full-text publications can be summarized by abstract sentences. We hypothesize that accessing images by abstract sentences is an improvement over the SummaryPlus

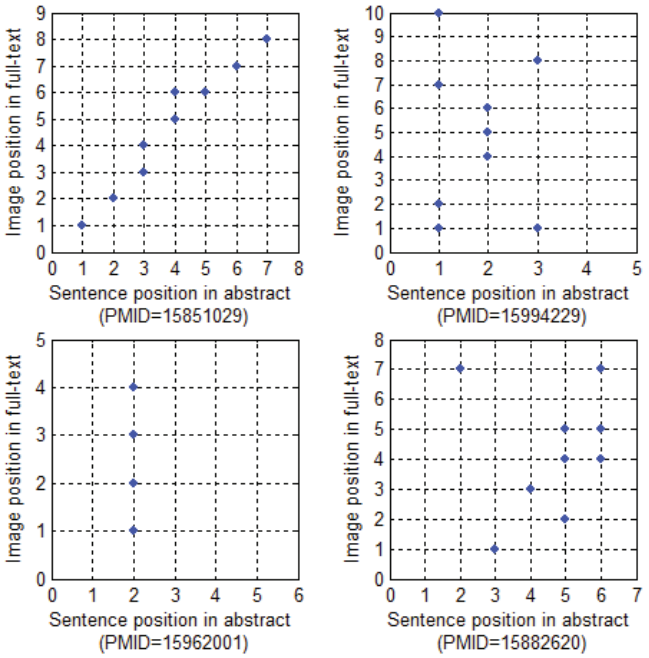


Fig. 3. Patterns that abstract sentences are linked to images in full-text articles. Numerical values in axes show the positions of abstract sentences or images. For example, ''2'' indicates the second abstract sentence or the second image that appear in the full-text article.

user-interface because the former will overcome the disadvantages of disjoint image content and may be the most efficient way to access images.

In order to evaluate whether biologists would prefer to accessing images from abstract sentence links, we have designed BioEx (Figure 4) and two other baseline user-interfaces (Figure 5). All three user-interfaces can be accessed at http://dmbi.columbia.edu/~yuh9001/BioEx.html. BioEx is built upon the PubMed user-interface except that images can be accessed by the abstract sentences. We have chosen the PubMed user-interface design because it has more than 70 million hits a month and represents the most familiar user-interface to biologists. Other information systems have also adapted the PubMed user-interface for similar reasons (Smalheiser and Swanson 1998; Hearst 2003). The two other baseline user-interfaces (as shown in Figure 5) were the original PubMed user-interface (Figure 5A) and a modified version of the SummaryPlus user-interface (Figure 5B), in which the images are listed as the disjointed thumbnails, rather than the links by abstract sentences.

We asked the 119 biologists who had linked sentences to images in their publications to assign a label to each of the three user-interfaces to be ''My favorite'', ''My second favorite'', or ''My least favorite''[1]. We designed the evaluation so that a user-interface's label is independent of the choices of the other two user-interfaces. A total of 41 or 34.5% of the biologists whom we requested completed the evaluation. Table 2 shows their choices. Table 2 shows that 36 or 87.8% of the total 41 biologists judged

[1]We assume BioEx is a useful improvement over the PubMed user-interface, a baseline that has already been favored by biologists.

Fig. 4. BioEx user-interface (as shown in A) is built upon the PubMed user-interface. Images are shown as thumbnails at the bottom of a PubMed abstract. When a mouse (as shown as a hand in A) moves to ''Fig x'', it shows the associated abstract sentence(s) that link to the original figure that appears in the full-text articles. For example, ''Fig 1'' links to image B. ''Related Text'' provides links to other associated text besides the image caption. The user-interface can be accessed from the link at http://dbmi.columbia.edu/~yuh9001/BioEx.html.

Fig. 5. Baseline user interfaces. (A): The PubMed interface of the article (PMID=15851029). (B): SummaryPlus in which images are listed as the thumbnails at the bottom of the abstract.

Table 2. Preferences made by 41 biologists who evaluated the three user-interfaces

	Favorite	Second Favorite	Least Favorite
PubMed	1	11	29
SummaryPlus	6	26	9
BioEx	36	3	2

Table 3. Comments made by biologists who evaluated BioEx and two other baseline user-interfaces

C1. C (i.e., BioEx) would be useful, because one can easily confirm the strength/validity of a sentence in the abstract. Sometimes I search abstracts looking for information on a specific question; this would be helpful to evaluate the abstracts. B (i.e., SummaryPlus) is not very useful, because random images are difficult to interpret.
C2. Adding links to the figures significantly facilitates more in depth skimming of the literature. Case B (i.e., SummaryPlus) is a significant improvement over case A (i.e., PubMed). Case C (i.e., BioEx) simplifies accessing the appropriate figures to evaluate the approaches used and is a useful improvement over case B.
C3. The second (i.e., SummaryPlus) permits accessing figures while retaining continuity of the abstract and remaining an economical extension and improvement to the existing PubMed system.
C4. Instead with thumbnails the links could be labeled with Fig.X. This would be more informative.

that BioEx is ''My favorite''. One biologist judged all three user-interfaces to be ''My favorite'' and 5 other biologists considered SummaryPlus as ''My favorite'', two of whom (or 4.9% of the total 41 biologists) judged BioEx to be ''My least favorite''. The SummaryPlus user-interface was the second choice by a majority of biologists (63.4%).

A total of eighteen biologists not only evaluated the three user-interfaces, but also provided us with additional text comments. Of those 18 biologists, 17 of them made positive comments regarding to BioEx and 3 of 17 additionally made suggestions to enhance the BioEx user-interface design[2]. Table 3 shows selected original comments made by the biologist evaluators: two are positive (C1∼C2); C3 is negative; and C4 shows a suggestion to enhance the BioEx interface design.

4 STRATEGIES FOR LINKING ABSTRACT SENTENCES TO IMAGES

One way to implement BioEx is to ask the authors of a paper to link abstract sentences to images. However, currently, PubMed has more than 15 million citations. It is not feasible to ask the authors to perform such a large scale of annotation, although it may be feasible for the publishers to request such a task when a new manuscript is accepted. In order to implement BioEx, we need to explore statistical and natural language processing approaches to automatically identify abstract sentences that summarize images.

[2]Our initial BioEx user-interface applied thumbnails at the end of abstract sentences. Three biologists had made suggestions to replace the thumbnails with ''Fig X''. Our current BioEx therefore is implemented based on such recommendations.

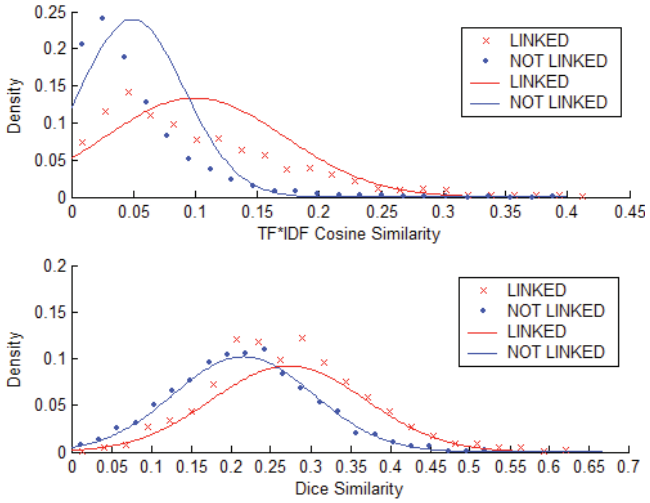


Fig. 6. The distribution of linked and unlinked abstract sentences or images as a function of TF*IDF weighted cosine similarity and Dice's coefficient.

We may simplify the task of linking abstract sentences to images as a task of aligning abstract sentences to other associated text (i.e., captions and other embedded text) that correspond to the same images. Such simplification is based on two assumptions. The first is that image content consistently corresponds to its associated text in the full-text articles; the correspondence is evidently supported by the work (Rafkind *et al*. 2006) that explored both text features and image features for biomedical literature image classification.

The second assumption is that there are strong word similarities between abstract sentences and other associated texts. To validate this assumption, we plotted the link distribution as a function of word similarity using the 114 annotated full-text articles. We examined two similarity measurements; namely, Dice's coefficiency (Dice 1945; van Rijsbergen 1979) and the TF*IDF weighted cosine coefficient (Witten *et al*. 1999), both of which are commonly used in tasks including information retrieval and topic detection. Dice score $D(i, j)$ was calculated by formula (1). The TF*IDF weighted cosine coefficient score (Salton and Lesk 1968; Witten *et al*. 1999) $sim(i, j)$ was calculated by (2) and (3). W_i and W_j are the total number of words in texts i and j, where i and j are either abstract sentences or image captions. $W_{ij} = W_i \cap W_j$. In (2) and (3), inverse document frequency (IDF) of a word is calculated from all sentences (N) in the full-text article.

$$D(i,j) = \frac{2W_{ij}}{W_i + W_j} \tag{1}$$

$$idf(w) = \log_{10}\left(\frac{N}{N(w)}\right) \tag{2}$$

$$\frac{sim(i,j) = \sum_w^{W_i \cup W_j} tf_i(w) * tf_j(w) * idf(w)}{\sqrt{\sum_w^{W_i \cup W_j} tf_i^2(w))} * \sqrt{\sum_w^{W_i \cup W_j} tf_j^2(w))}} \tag{3}$$

Figure 6 shows that both TF*IDF weighted cosine similarity and Dice score can separate linked pairs from unlinked pairs and the

TF*IDF weighted cosine similarity shows an advance over the Dice's score for separating the two. The results empirically validate our assumption that there are word similarities between abstract sentences to their corresponding image captions.

We may explore different models to map abstract sentences to images. For example, linking abstract sentences to image captions and other associated text can be treated as a task of sentence alignment in machine translation. However, we consider that the former is a more challenging task than the latter. In machine translation, most of the sentences are aligned and typically a majority of sentences are aligned one to one (i.e., one sentence is translated to only one sentence in the second language). For example, in (Gale and Church 1993), 89% sentences belonged to this category. However, in our data collection, many abstract sentences and images do not have any corresponding images or sentences and many abstract sentences and images correspond to two or more images and abstract sentences (details in Table 1).

Furthermore, techniques that were successful in machine translation might not apply to our task of linking abstract sentences to images. For example, sentence length (i.e., a long sentence must be translated to a long sentence in another language) was found to be powerful in sentence alignment. However, we do not find direct correspondence between the length of an abstract sentence and the length of the corresponding image caption. Additionally, in machine translation, most of the sentences were aligned in the order they appear. However, orderly alignment does not apply to many cases in our data collection (examples shown in Figure 3B–D).

We therefore explored a model that applies hierarchical clustering algorithms to cluster abstract sentences and images based on word similarities which has shown in Figure 6 to be able to separate linked abstract image pairs from unlinked ones. In our application, if abstract sentences belong to the same cluster that includes images, the abstract sentences summarize the image content. The clustering model holds advantages over other models in that the clustering methods flexibly allow ''one-to-many'' and ''many-to-one'' mapping. Furthermore, we will show later (Section 5.3) that it is a relatively a simple task to incorporate positional information.

5 APPLYING HIERARCHICAL CLUSTERING ALGORITHM FOR AUTOMATICALLY LINKING ABSTRACT SENTENCES WITH IMAGES

Hierarchical clustering algorithms are well-established algorithms that are widely used in many other research areas including biological sequence alignment (Corpet 1988), gene expression analyses (Herrero *et al*. 2001), and topic detection (Lee *et al*. 2006). The algorithm starts with a set of text (i.e., abstract sentences or image captions). Each sentence or image caption represents a document that needs to be clustered. The algorithm identifies pair-wise document similarity and then merges the two documents with the highest similarity into one cluster. It then re-evaluates pairs of documents/clusters; two clusters can be merged if the average similarity across all pairs of documents within the two clusters exceeds a predefined threshold. In presence of multiple clusters that can be merged at any time, the pair of clusters with the highest similarity is always preferred. See (Lee *et al*. 2006) for a detailed description and evaluation of the algorithm. We calculated pairwise document similarity based on the TF*IDF weighted cosine similarity. We had previously shown that the TF*IDF method shows advance over the Dice method (Figure 6). We explored different word features, weights, positional information, and clustering strategies.

5.1 Word features

We have explored bag-of-words and n-grams as features for the clustering tasks. Additionally, we have explored different feature combinations that include features in caption, other associated text, neighboring text, synonyms, or combined.

(1) **Caption** An image caption usually incorporates multiple sentences or phrases. The heading usually provides an abstraction of the entire image content and the first sentence of each subheading provides a summary of each sub-experiment. We have explored the combinations of the heading and the first sentences of the subheading. Specifically, we explored 1) all words in the caption, 2) heading plus the first sentence of each sub-experiment in the image caption, and 3) the first sentence of each sub-experiment.

(2) **Other Associated Text** The image caption is not the only content that describes the experiment. There is other associated text in the full-text document that may provide additional discriminating features for clustering. We have identified this ''other associated text'' by surface cues: we extract paragraphs incorporating ''Figure X'' from the full-text article, then merge these paragraphs with the corresponding image captions and subject the merged text to the clustering procedure. Our approach stems from the fact that biologists frequently devote an entire paragraph or more to describing the results of one experiment.

(3) **Neighbouring Text** Abstract sentences are coherent and the neighbouring sentences (the preceding and the following sentences) may be content-related. Furthermore, we found that 135 out of the total 746 images or 18% images in our data collection correspond to consecutive abstract sentences. For example, Figure 3 shows that the two abstract sentences

> ''a purified Rae1 complex stabilizes microtubules in egg extracts in a RanGTP/importin beta-regulated manner''

and

> ''interestingly, Rae1 exists in a large ribonucleoprotein complex, which requires RNA for its activity to control microtubule dynamics in vitro''

point to the same image ''Fig 6''. We therefore explored ''neighbouring text'' as additional features: we merged the features of the neighbouring abstract sentences, namely, the previous and the following sentences, with the abstract sentence to be examined and applied the merged features to identify images that are associated with the abstract sentence.

(4) **Synonym Expansion** Abstract sentences and image captions do not always use the exact same words. Synonym expansion might enhance the clustering performance. We applied the large biomedical knowledge resource the Unified Medical Language System (UMLS) (Humphreys *et al*. 1998) to expand synonyms. The UMLS incorporates more than one million biomedical concepts with synonyms. We applied a simple

string matching to capture the terms and to map terms to the UMLS concepts and synonyms.

5.2 Word weight

For document clustering, we applied the TF*IDF weighted cosine similarity, which was shown in the previous section 4 to have a better discrimination than the Dice's score. We treat each sentence or image caption as a ''document'' and the features are bag-of-words. We explored three different methods to obtain the TF*IDF value for each word feature:

(1) ***IDF(abstract+caption)***: the IDF values were calculated from the pool of abstract sentences and image captions;

(2) ***IDF(full-text)***: the IDF values were calculated from all sentences in the full-text article;

(3) ***IDF(abstract)***::***IDF(caption)***: we obtained two sets of IDF values. For words that appear in abstracts, the IDF values were calculated from the abstract sentences; for words that appear in image captions, the IDF values were calculated from the image captions.

5.3 Position

Although we show that in many of the annotated full-text articles, the abstract sentences do not correspond to images in the order they appear in the full-text articles (examples shown in Figure 3B∼D), we found that the chance that two abstract sentences or images link to an image or an abstract sentence decreases when the distance between two abstract sentences or images increases. For example, two consecutive abstract sentences have a higher probability to link to one image than two abstract sentences that are far apart. Such ''positional distance'' also applies to images: two consecutive images have a higher chance to link to the same abstract sentence than two images that are separated by many other images. To integrate such positional information into our existing hierarchical clustering algorithms, we modified the TF*IDF weighted cosine similarity with positional distance. Assuming that we consider an abstract sentence or an image caption as a document, the TF*IDF weighted cosine similarity for a pair of document i and j is $sim(i,j)$, we integrated the positional distance, and the final similarity $SIM(i,j)$ is:

$$SIM(i,j) = sim(i,j) * \left(1 - abs\left(\frac{P_i}{T_i} - \frac{P_j}{T_j}\right)\right) \qquad (4)$$

(1) If i and j are both abstract sentences, T_i=T_j=*total number of abstract sentences*; and P_i and P_j represents the positions of sentences i and j in the abstract.

(2) If i and j are both image captions, Ti=Tj=total number of images that appear in a full-text article; and Pi and Pj represents the positions of images i and j in the full-text article.

(3) If i and j are an abstract sentence and an image caption, respectively, Ti=total number of abstract sentences and Tj=total number of images that appear in a full-text article; and Pi and Pj represent the positions of abstract sentence i and image j.

5.4 Clustering strategy

Although there are a great deal of word similarities between abstract sentences and their corresponding image captions, there are also significant differences between the two texts. In general, image captions tend to be long and incorporate content-lean experimental details. For example, the image caption (Fig 1) in Figure 3 is

> ''(A) Schematic of the assay used to identify Rae1. Sequential affinity chromatography steps were used to deplete metaphase-arrested CSF Xenopus egg extracts: first, a RanGTP matrix was used to remove RanGTP binding proteins including importin β (ΔRanBP Extract), freeing cargoes that caused spontaneous microtubule aster formation. . . .'',

which is in contrast to its succinct abstract sentence:

> ''Here, we have used an activity-based assay in Xenopus egg extracts to purify the mRNA export protein Rae1 as a spindle assembly factor regulated by this pathway''.

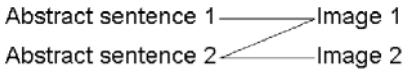
To best capture the differences between abstract sentences and image captions, we explored three clustering strategies; namely, per-image, per-abstract sentence, and mix.

(1) **Per-image** clusters each image caption with all abstract sentences. The image is assigned to (an) abstract sentence(s) if they belong to the same cluster. This method values features in abstract sentences more than image captions because the decision that an image belongs to (a) sentence(s) depends upon the features from all abstract sentences and the examined image caption. The features from other image captions will not play a role for the clustering.

(2) **Per-abstract-sentence** takes each abstract sentence and clusters it with all image captions that appear in a full-text article. Images are assigned to the sentence if they belong to the same cluster. This method values features in image captions higher than the features in abstract sentences because the decision that an abstract sentence belongs to image(s) depends upon the features from the image captions and the examined abstract sentence. The features from other abstract sentences will not play a role for the clustering.

(3) **Mix** clusters all image captions with all abstract sentences. This method treats features in abstract sentences and image captions equally.

In addition, because the clusters generated by the hierarchical clustering algorithms are typically mutually exclusive, **Mix** will never achieve 100% accuracy for detecting the following links:



If grouping into two clusters (abstract_sent_1, image_1) and (abstract_sent_2, image_2), **Mix** will miss the link between abstract_sent_2 and image_1; if grouping into two clusters (abstract_sent_1, image1, abstract_sent_2) and (image_2), **Mix** will miss the link between abstract_sent_2 and image 2; if grouping into two clusters (abstract_sent_2, image 1, image 2) and (abstract_sent_1), **Mix** will miss the link between abstract_sent_1 and image_1. Finally, if grouping into one cluster, **Mix** will create

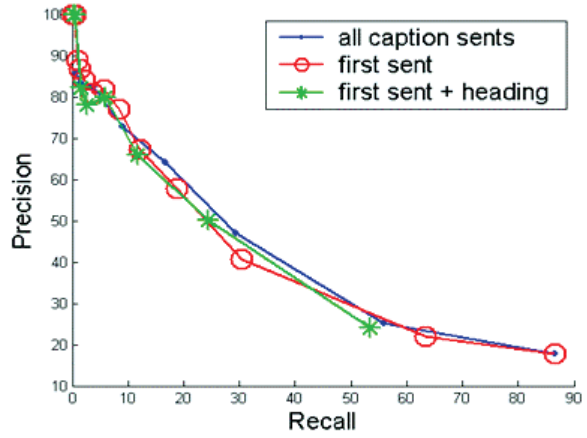


Fig. 7.

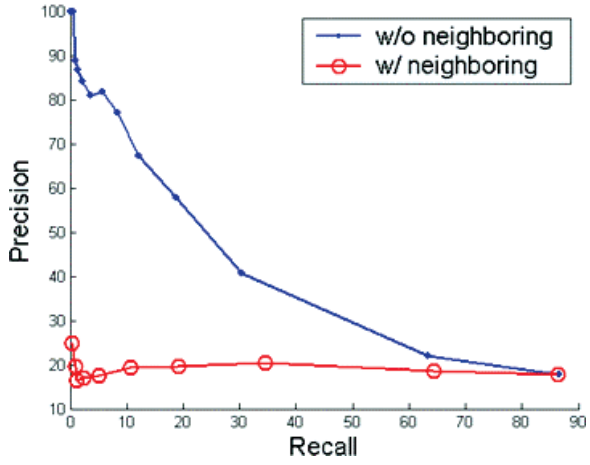


Fig. 9.

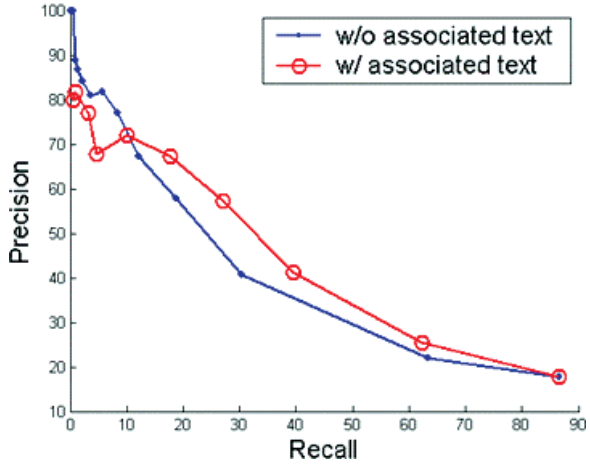


Fig. 8.

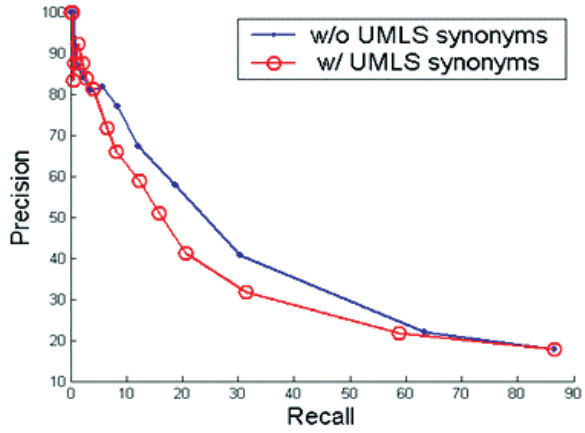


Fig. 10.

one false positive and if grouping into four clusters, **Mix** will generate three false negatives.

6 DATA AND EVALUATION METRICS

The 114 bioscience articles we described previously (Section 2) were used to evaluate the mapping between abstract sentences and images. We report *recall* and *precision* as the evaluation metrics for linking sentences to images. *Recall* is the total number of correctly predicted links divided by the total number of annotated links. *Precision* is the total number of correctly predicted links divided by the total number of predicted links.

7 RESULTS AND DISCUSSION

Figures 7–13 show the results in which we explored different combinations of features and algorithms. The default parameters for all these experiments were ''per image'', ''without UMLS synonyms'', ''bag-of-words'', and "IDF(abstract_caption), ''without neighboring sentences'' and ''without position''.

Figure 7 shows the results in which we explored image captions, the combined heading with the first sentence from each sub-experiment, and the first sentence from each sub-experiment. The results show that incorporating all image captions as features leads to a slightly better performance over the other features.

Figure 8 shows that the clustering performance increases when features include other associated text. The results directly support our assumptions that other associated text represents images content and that there are lexical similarities between abstract sentence and other associated text that correspond to an image. Because the feature spaces have been expanded, the overall recall and precision have increased. On the other hand, the high-end precision has dropped from 100% to 80%. This can be explained by the fact that although other associated text may incorporate useful word features that do not appear in captions, they may also include words that never appear in the corresponding abstract sentences, and those words introduce ''noise'' at the clustering. Additionally, we currently implemented a simple approach for identifying other associated text: we identified the entire paragraph as the ''other associated text'' if the paragraph contains the surface cue ''Figure X''. The approach will introduce significant ''noise''

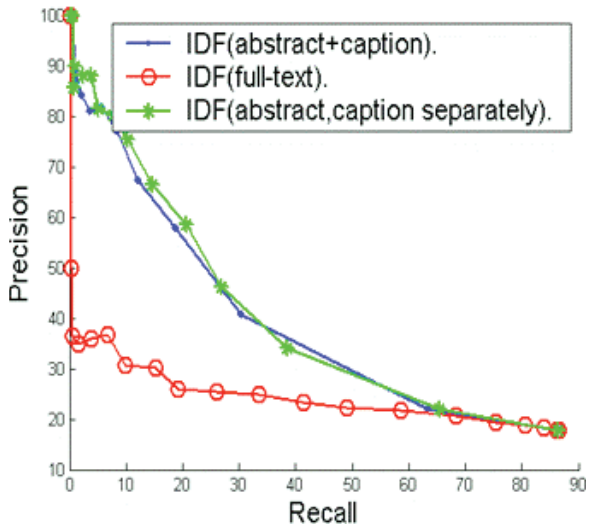


Fig. 11.

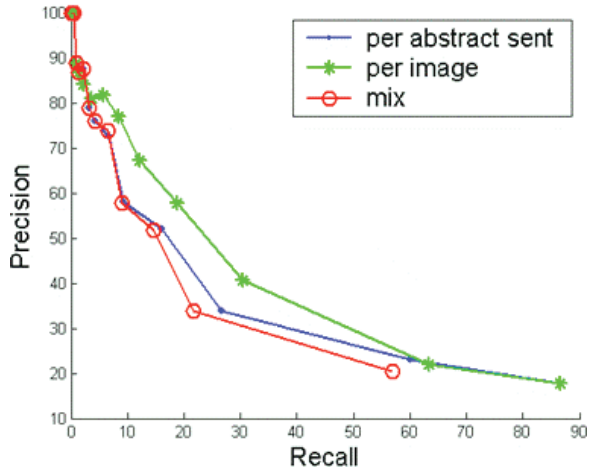


Fig. 12.

because frequently, a paragraph may describe more than one experiment.

Figure 9 shows that ''without neighboring sentences'' greatly outperformed ''with neighboring''. Recall that neighboring sentences are adjacent sentences (or proceeding and following sentences) of the examined sentence. The results conclude that the ''useful'' information introduced by the neighboring sentences is overshadowed by the ''noise''. The results are not entirely surprising. Although 18% images in our data collection correspond to consecutive abstract sentences, we found that a majority of images do not. Specifically, 424 (57.1%) images correspond to single abstract sentences, 91 (12.3%) images correspond to non-consecutive abstract sentences, and 92 (12.4%) images do not link to any of abstract sentences.

Figure 10 shows that synonym expansion has a disappointing performance. The results may contribute to several factors, including how robust was the mapping between a string to the UMLS concepts and the problems of homonyms. We will describe in the next section (Section 8 Future Work) how we will explore different approaches to enhance synonymous term identification.

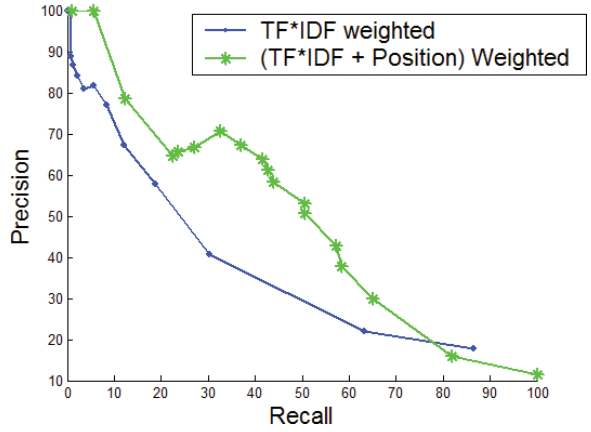


Fig. 13.

Our results also show little performance differences between unigram and n-grams (data not shown). The results are not surprising because of the problem of data sparseness. Many other natural language processing systems have found little gain of n-gram in either topic detection (Lee *et al*. 2006) or document and sentence classification (Yu and Hatzivassiloglou 2003).

Figure 11 shows the performance of three different methods for calculating the IDF values. The results show that the ''global'' IDFs, or the IDFs obtained from the full-text article, has a much lower performance than ''local'' IDFs, or IDFs calculated from the abstract sentences and image captions. The results suggest that abstract sentences and image captions alone are more accurate than the whole full-text article for estimating the importance of features in our task of linking abstract sentences to image captions. In addition, IDFs that were separately calculated from the abstract sentences and image captions performs slightly better than the combined IDFs. The results suggest that the distributions of features are different between abstract sentences and image captions.

Recall that we have explored three strategies for linking abstract sentences to images; namely, **Per-image** that takes each image caption and clusters it with abstract sentences, **Per-abstract-sentence** that takes each abstract sentence and clusters it with image captions, and **Mix** that clusters all image captions with all abstract sentences. As we have predicted, Figure 12 shows that both **Per-image** and **Per-abstract-sentence** out-performs Mix. Furthermore, **Per-image** significantly out-performs **Per-abstract-sentence**. The results suggest that features in abstract sentences are more useful than features in caption for the task of clustering.

Figure 13 shows that combining word features with position has significantly enhanced the performance. When the recall is 33%, the precision of combining TF*IDF with positional information increases to 72% from the original 38%, which corresponds to a 34% absolute increase. The results strongly indicate the importance of positional information. When the precision is 100%, the recall is 4.6%. We believe that a high precision is the key to success for this application. Many previous successful and applicable natural language processing systems have also achieved high precisions (e.g., (Friedman *et al*. 2001)). However, the low recall will render

Table 4. The recall values for different types of links.

Type	Recall
1:1	32.5%
1:2	49.3%
1:3	36.5%
1:4	30.8%
1:5	24.4%
1:6	29.2%
1:7	57.1%
2:1	44.2%
3:1	39.8%
4:1	35.7%
5:1	50.0%

our current system's application for the real application. We have implemented BioEx (with a recall of 33% and a precision of 72%) that can be accessed at http://dbmi.columbia.edu/~yuh9001/BioEx.html, from which a user can query 17,000 downloaded full-text Proceedings of the National Academy of Sciences (PNAS) full-text articles.

Recall that our evaluation data consists of three types of mapping between abstract sentences and images. They are *one to one*, *one to many* and *many to one*. Previous dynamic programming methods in machine translation had showed significant decreases in performance when a sentence was aligned to multiple sentences (Gale and Church 1993). We therefore examined the performance of our algorithms for each type. Since we could not measure the precision for this task because we miss the false positives for each type, we compared the recall for different type (results shown in Table 4). We chose the system with the overall f-score=44.4% (F-score=2*recall*precision/(recall+precision)). Our results, in contrast, do not show significant differences in recall among three types of mapping. Our results may support the robustness and advantages of the hierarchical clustering methods over the dynamic programming method for this application.

8 FUTURE WORK

Our current evaluation data were annotated by biologists who are the authors of their publications. We have observed inconsistency in annotation. For example, by manually examining 114 annotated full-text articles, we found that many biologists assigned images to conclusions and speculations, while others did not. For example, the last sentence in the abstract (pmid=15933717) is:

> ''Taken together, our observations suggest that PIASy is a critical regulator of mitotic SUMO-2 conjugation for Topoisomerase-II and other chromosomal substrates, and that its activity may have particular relevance for centromeric functions required for proper chromosome segregation.''

which was assigned to all five images that appeared in the full-text article. On the other hand, in two other articles (Figure 3A and D), the authors did not assign any images to their corresponding conclusions and speculations. We believe that such inconsistency may be fixed in the future if with a carefully designed annotation instruction. Additionally, future work one may need to measure the inter-rater reliability for linking abstract sentences to images; this requires that one article must be annotated by two or more people. One way is to ask the co-authors to annotate their articles independently and then measure the agreement among them.

We believe that there are rooms to enhance the performance for linking abstract sentences to images. In this study, we applied word-level similarity to measure the link between abstract sentences to images. However, exact word matching may not be the best solution in this application. One may need to capture synonymous terms. For example, if we could capture the abbreviation ''NMR'' and map it to the corresponding full form ''nuclear magnetic resonance'', the clustering algorithm would be able to link the two texts

> ''Here we report nuclear magnetic resonance and X-ray protein structures of the N-terminal substrate recognition domain of FimD (FimDN) before and after binding of a chaperoné C subunit complex'' (an abstract sentence)

and

> ''NMR studies on FimD N...'' (an image caption; pmid= 15920478).

One may explore the work of (Aronson 2001) that applied the large biomedical knowledge resource the Unified Medical Language System (UMLS) for synonym identification and the work of (Yu *et al*. 2002) that explored rule-based approach for capturing abbreviations and full forms from literature.

Additionally, it may also be important to capture semantic similar terms. For example, if we link ''Death'' to ''toxicity,'' we could recover the link between the following two statements:

> ''Acute and chronic exposure to kainate caused extensive oligodendrocyte death in culture'' (an abstract sentence) and

> ''Kainate toxicity in oligodendrocytes derived from P7 rat optic nerves'' (an image caption; pmid= 9238063).

Currently, our system missed the links. For identifying semantically related terms, one may explore the work of (Lin 1998a; Lin 1998b; Yu and Agichtein 2003). Although we explored hierarchical clustering methods in this study and had shown the advantages of these methods. Future work one may explore dynamic programming that has been successful for many other tasks including sequence alignment (Lawrence *et al*. 1993), gene or protein name recognition (Krauthammer *et al*. 2000), paraphrasing (Barzilay and Lee 2003), and sentence alignment (Chen 1993). The current algorithm does not consider the ''ordering effect'', which is that the position order of image pairs (i.e., one image appears ahead of the other image) reflects the position order of abstract sentences or the abstract sentence(s) appear with the same order of corresponding images. Although such alignment does not apply to every full-text article, we found that out of the total of 1649 image pairs in our 114 annotated full-text articles, 1207 or 73.6% image pairs appear with the same order of their corresponding sentences. Dynamic programming methods had shown to be powerful for detecting such alignment.

9 CONCLUSION

As described in this paper, we have designed and evaluated a novel user-interface BioEx that allows biologists to directly access images

by abstract sentences. Current, more than 40 biologists evaluated the BioEx user-interface and 87.8% of them were in favor of BioEx over two other baseline systems. Additionally, we have also explored natural language processing approaches, specifically, the hierarchical clustering algorithms, to automatically link abstract sentences to images. We have explored different features and algorithms. One of the best systems shows a performance of 100% precision with 4.6% recall. We believe a high precision is a key to success for this application, although BioEx may not be applicable to real use at the current stage. We have implemented BioEx (with a recall of 33% and a precision of 72%) that can be accessed from the link at http://dbmi.columbia.edu/~yuh9001/BioEx.html, from which biologists can query 17,000 downloaded Proceedings of the National Academy of Sciences (PNAS) full-text articles.

ACKNOWLEDGEMENTS

The authors thank 119 biologists who had annotated data for us and 41 biologists who had evaluated the BioEx user-interface. The list of biologists is available upon request. The authors in particular thank Dr. Weiqing Wang for her contribution to this work. As a biologist, Dr. Wang had helped us identified the information needs by biologists. Dr. Wang had additionally helped us by collecting annotations from biologists and by commenting and evaluating BioEx user-interfaces. The authors are partially supported by JDRF 6-2005-835. Any opinions, findings, or recommendations are those of the authors and do not necessarily reflect JDRF's views.

REFERENCES

Alfarano C, Andrade CE, Anthony K, Bahroos N, Bajec M, Bantoft K, Betel D, et al. (2005) The Biomolecular Interaction Network Database and related tools 2005 update. Nucleic Acids Res 33:D418–424.

Aronson A (2001) Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. Paper presented at American Medical Information Association.

Barzilay R, Lee L (2003) Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. Paper presented at NAACL-HLT.

Chen SF (1993) Aligning sentences in bilingual corpora using lexical information. Paper presented at The 31st annual meeting on Association for Computational Linguistics.

Corpet F (1988) Multiple sequence alignment with hierarchical clustering. Nucleic Acids Res 16:10881–10890.

Dice LR (1945) Measures of the amount of ecologic association between species. Journal of Ecology 26:297–302.

Gale W, Church K (1993) A program for aligning sentences in bilingual corpora. Computational Linguistics 19:75–102.

Gross A (2002) Communicating science: the scientific article from the 17th century to the present. Oxford University Press, New York.

Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA (2005) Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic Acids Res 33:D514–517.

Hearst M (2003) The BioText project. A powerpoint presentation.

Herrero J, Valencia A, Dopazo J (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics 17:126–136.

Hoffmann R, Valencia A (2005) Implementing the iHOP concept for navigation of biomedical literature. Bioinformatics 21 Suppl 2:ii252-ii258.

Humphreys BL, Lindberg DA, Schoolman HM, Barnett GO (1998) The Unified Medical Language System: an informatics research collaboration. J Am Med Inform Assoc 5:1–11.

Jensen LJ, Saric J, Bork P (2006) Literature mining for the biologist: from information retrieval to biological discovery. Nat Rev Genet 7:119–129.

Krauthammer M, Rzhetsky A, Morozov P, Friedman C (2000) Using BLAST for identifying gene and protein names in journal articles. Gene 259:245–252.

Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. 8:208–214.

Lee M, Wang W, Yu H (2006) Exploring supervised and unsupervised methods to detect topics in Biomedical text. BMC Bioinformatics 7:140.

Lin DK (1998a) Automatic retrieval and clustering of similar words. Paper presented at Proceedings of ACL-98. Pittsburg, Pennsylvania.

Lin DK (1998b) An information-theoretic definition of similarity. Paper presented at Int Conf on Machine Learning.

Mosteller F, Wallace D (1963) Inference in an authorship problem. Journal of the American Statistical Association 58:275–309.

Rafkind B, Lee M, Chang S, Yu H (2006) Exploring text and image features to classify images in bioscience literature. Paper presented at HLT-NAACL Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis.

Rzhetsky A, Iossifov I, Koike T, Krauthammer M, Kra P, Morris M, Yu H, Duboue PA, Weng W, Wilbur WJ, Hatzivassiloglou V, Friedman C (2004) GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. J Biomed Inform 37:43–53.

Salton G, Lesk ME (1968) Computer evaluation of indexing and text processing. J ACM 15:8–36.

Smalheiser NR, Swanson DR (1998) Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses. Comput Methods Programs Biomed 57:149–153.

van Rijsbergen CJ (1979) Information Retrieval. Butterworths.

Witten I, Moffat A, Bell T (1999) Managing Gigabytes: Compressing and indexing documents and images. Morgan Kaufmann Publishers.

Yu H, Agichtein E (2003) Extracting synonymous gene and protein terms from biological literature. Bioinformatics 19 Suppl 1:i340–349.

Yu H, Hatzivassiloglou V (2003) Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. Paper presented at Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP 2003).

Yu H, Hripcsak G, Friedman C (2002) Mapping abbreviations to full forms in biomedical articles. J Am Med Inform Assoc 9:262–272.

Yu H, Lee M, Ely J, Osheroff J, Cimino J (2006) Beyond information retrieval: Medical question answering. Journal of Biomedical Informatics.

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e557–e565
doi:10.1093/bioinformatics/btl232

A sequence-based filtering method for ncRNA identification and its application to searching for riboswitch elements

Shaojie Zhang[1,*], Ilya Borovok[2], Yair Aharonowitz[2], Roded Sharan[3] and Vineet Bafna[1]

[1]Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA, [2]Department of Molecular Microbiology and Biotechnology and [3]School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel

ABSTRACT

Motivation: Recent studies have uncovered an "RNA world", in which non coding RNA (ncRNA) sequences play a central role in the regulation of gene expression. Computational studies on ncRNA have been directed toward developing detection methods for ncRNAs. State-of-the-art methods for the problem, like covariance models, suffer from high computational cost, underscoring the need for efficient filtering approaches that can identify promising sequence segments and speedup the detection process.

Results: In this paper we make several contributions toward this goal. First, we formalize the concept of a filter and provide figures of merit that allow comparison between filters. Second, we design efficient sequence based filters that dominate the current state-of-the-art HMM filters. Third, we provide a new formulation of the covariance model that allows speeding up RNA alignment. We demonstrate the power of our approach on both synthetic data and real bacterial genomes. We then apply our algorithm to the detection of novel riboswitch elements from the whole bacterial and archaeal genomes. Our results point to a number of novel riboswitch candidates, and include genomes that were not previously known to contain riboswitches.

Availability: The program is available upon request from the authors.

Contact: shzhang@cs.ucsd.edu

1 INTRODUCTION

A *database filter* is a computational procedure that takes a database as input, and outputs a subset of the database. The goal is to ensure that the object being searched for remains in the database after filtering, the filtered database is significantly smaller, and the filtering operation is very fast. Filters have played a central role in bioinformatics. BLAST is the prototypical example, with a keyword match filter greatly improving the search for remote homologs. Indeed, improving the filters for sequence similarity search remains an intensively researched area, with many recent publications. Filtering is also being applied in other bioinformatics domains, including structural genomics (Leibowitz *et al*., 1999), proteomics (mass-spectrometry) (Frank *et al*., 2005; Tanner *et al*., 2005), and non coding RNA (ncRNA) (Weinberg and Ruzzo, 2004a,b; Zhang *et al*., 2005). Here, we revisit the notion of filtering, focusing on applications to detecting ncRNAs.

ncRNAs are genomic sequences that are transcribed, but not translated, and function as RNA molecules. Recent discoveries of many novel families and sub-families of ncRNA have underscored their importance, and hint at an RNA world, where coding and non-coding genes play equally important roles (Eddy, 2001; Storz, 2002; Vitreschak *et al*., 2004). The signal for ncRNA is considerably weaker than that for protein coding genes and *de novo* approaches that look for secondary structure or transcriptional initializing signal do not work well (Rivas and Eddy, 2000). Therefore, comparative approaches are more popular with two major directions. One way is to look at compensatory mutations (or consensus folds) in pre-aligned orthologous regions (Rivas and Eddy, 2001; Washietl *et al*., 2005; Pedersen *et al*., 2006). However, the success of this method relies on a good ''structural'' alignment which is difficult to get (Bafna *et al*., 2006). The other comparative approach to discovering novel homologs of a query ncRNA is also increasing in importance, much like BLAST is often used to identify novel homologs of coding genes. While viable, this approach poses a technical challenge since the known algorithms for aligning ncRNA are at least an order of magnitude slower than sequence alignment (Klein and Eddy, 2003; Zhang *et al*., 2005), and even slower when other secondary structures (such as pseudoknots) are allowed (Dost *et al*., 2006). Indeed, using a search based on a covariance model (CM) (Durbin *et al*., 1998), it would take 54 hours to query two bacterial genomes: *E. coli* K12 and *Staphylococcus aureus* MW2 (7.5 Mb) for a sub-family such as the FMN riboswitch (145 bp). This makes the filtering problem both easier and harder. On the one hand, the alignment is so expensive (cubic time), that even a computationally intensive filter (quadratic time) could be useful. At the same time, since the alignment is so expensive, the filtering itself must be very *efficient* in removing a large portion of the database while retaining the true hits. For example, a filter that removes 50% of the database is still not sufficient to make CM searches tractable for large genomic sequences.

Algorithms that align ncRNA are expensive because they score for both sequence and structure conservation, and the latter task is computationally intensive. Filtering for RNA was systematically explored by Weinberg and Ruzzo (2004a, b) who used a pigeonhole argument to show that it is enough to scan for sequence similarity,

*To whom correspondence should be adderessed.

expressed by a hidden Markov model, leaving the more expensive structural alignment for the filtered sequence. Henceforth, we refer to their filter as *HMM-filter*. Subsequently, they, and independently, some of us also used partial structure conservation for the filtering (Weinberg and Ruzzo, 2004b; Zhang *et al*., 2005). Even after applying these filters, the problem remains computationally expensive, and it is worthwhile to ask if one can do better.

Here, we make several contributions in this regard. First, we formalize the concept of a filter and provide figures of merit that allow comparison between filters. Second, we design novel filters and show that they dominate the HMM filters of Weinberg and Ruzzo (2004a) (we defer a formal definition of the notion of notion of dominance to Section 2). In practice, this leads to 1-2 orders of magnitude decrease in search time. However, our main point is not that we can build better filters, but that it is relatively easy to do so. Indeed, the filters we design are very simple conceptually, indicating perhaps that we have only scratched the surface on this problem. The main contribution of this paper is a principled approach to combining filters that have different performance characteristics to achieve dominance (Section 3).

We also revisit the issue of alignment by aligning an RNA-profile to a filtered substring. We emphasize that there is a strong (practically, 1-1) correspondence with CMs in both the alignment algorithm, and the observed results. Indeed, the advantage of the CMs is that their parameters can be trained using the same formalization. However, our reformulation helps us take advantage of simple tricks like banding and others which help speed up the alignment without appreciable loss in accuracy (Section 4). Similar extensions would require a departure from the formalism of stochastic context free grammars that support CMs. This also has an impact on filtering. Unlike previous approaches, we do not tie the accuracy of our filtering procedure to the accuracy of an existing alignment procedure. Thus, it is relatively easy to use our filtering procedure in conjunction with other different alignment algorithms. For example, in recent work, we used the filtering to search genomes for pseudoknotted RNA (Dost *et al*., 2006).

Within ncRNA, we focus our attention on Riboswitches. Riboswitches are ncRNA elements that often occur in the 5' Untranslated Region (UTR) regions of genes (Mandal *et al*., 2004; Nahvi *et al*., 2003; Rodionov *et al*., 2003a; Sudarsan *et al*., 2003; Vitreschak *et al*., 2003, 2004). The riboswitches have a mode of action that one normally associates with proteins: they directly sense the levels of specific metabolites with a structurally conserved aptamer domain to regulate expression of downstream genes. Riboswitches respond to a wide range of metabolites including coenzymes, purines, amino acids and some others. Most riboswitches are predicted to be within UTRs of mRNAs that encode biosynthetic enzymes or metabolite and metal transporters. Novel members are continuously being discovered. The Rfam database (Griffiths-Jones *et al*., 2005), version 7.0, has members from 12 sub-families of riboswitches. Due to their widespread and exclusive occurrence in bacteria, they are attractive antimicrobial targets. Our results point to a number of novel candidates for each of these sub-families, and include genomes that were not previously known to contain riboswitches.

2 FORMALIZING NCRNA FILTERS

Covariance Models (CMs) are probabilistic context-free grammar models that describe both structure and sequence information of an RNA family (Durbin *et al*., 1998; Eddy, 2002). The score of an RNA sequence t against a CM model $\mathcal{M}$ is roughly the sum of two components: its sequence similarity to the modeled family, measured using a position specific scoring matrix (PSSM) of nucleotides, and its structural similarity, measured against the distribution of nucleotide pairs in aligned positions. Formally,

$$S(\mathcal{M},t) \sim \textsc{SeqScore}(\mathcal{M},t) + \textsc{StructScore}(\mathcal{M},t)$$

where SeqScrcore is the score of the PSSM part of $\mathcal{M}$ against t. For ungapped alignments, this would simply be the sum over all columns

$$\textsc{SeqScore}(\mathcal{M},t) = \sum_j \textsc{SeqScore}(\mathcal{M}_j, t_j).$$

If gaps are allowed, we must compute an alignment that optimizes $S[\mathcal{M}, t]$. The SeqScore computation is an order of magnitude faster than an optimum StructScore computation. Weinberg and Ruzzo (2004a) use this as the basis of their sequence based HMM filter[1]. For a given threshold T for $\mathcal{M}$, they compute a threshold T_{ps} as

$$T_{ps} = \min\{\textsc{SeqScore}(\mathcal{M},t) : S(\mathcal{M},t) \geq T\}.$$

This choice of T_{ps} ensures that each 'true homolog' ($S(\mathcal{M}, t) \geq T$) will pass the filter. Moreover, much of the database will be rejected by this filter, and will not undergo the more expensive CM alignment.

In order to improve upon this filter, we start with formalizing the definitions of a filter and its quality. A *filter F* takes a sequence as input and outputs sub-sequences. We assume the operating parameters (such as a threshold) as part of the filter definition. To make the notion of performance independent of the database, we measure it on a suitably defined random database sequence D, with a set of true sequences A embedded in D. The performance of the filter is measured with the following:

(1) **Running Time:** The running time $T_F(|D|, n)$ is a function of query length n, and database length $|D|$.

(2) **Efficiency:** Let $O_F(D)$ be the output of filter F. Define efficiency as $e_F = \frac{|O_F(D)|}{|D|}$. The lower the better.

(3) **Accuracy:** Let A_F denote the subset of true sequences that are accepted by the filter. Then accuracy is defined as $A_F = \frac{|A_F|}{|A|}$. The higher the better.

Filter F_1 *dominates* F_2 if it is faster, more accurate, and more efficient than F_2. Often, filters perform well in one or two but not all of these aspects. In many cases, they can be combined for further improvement. The two obvious ways to combine filters are:

- **Union F_1+F_2**: in which $O_{F_1+F_2}(D) = O_{F_1}(D) \cup O_{F_2}(D)$. Union helps if both F_1 and F_2 are fast and efficient, but not accurate.
- **Composition F_1·F_2**: $O_{F_1 \cdot F_2}(D) = O_{F_2}(O_{F_1}(D))$. Composition helps when the two filters are accurate but not very efficient,

[1]They use HMMs (not PSSMs) to describe the filter, but that technical difference does not change the argument.

and F_1 is faster than F_2. Note that composition is always better than intersection, as the running time $T_{F_1}(D,n) + T_{F_2}(O_{F_1}(D),n)$ is better than T_{F_2} with identical accuracy.

We will use both of these operations in designing better filters. The following result shows that it is not essential to be able to compute efficiency directly in order to prove dominance.

THEOREM 1. *Filter F can be dominated if there exists a filter F_1 with $A_F \subseteq A_{F_1}$ and $T_{F_1}(D,n)/T_F(D,n) \leq 1 - e_{F_1}$.*

PROOF. We simply use the composition $F_1 \cdot F$ as the filter. Clearly, it has better accuracy and is more efficient. For running time, we note that

$$\begin{aligned} T_{F_1}(D,n) + T_F(O_{F_1}(D),n) &\leq T_{F_1}(D,n) + e_{F_1}T_F(D,n) \\ &\leq (1 - e_{F_1})T_F(D,n) + e_{F_1}T_F(D,n) \\ &\leq T_F(D,n). \end{aligned}$$

While self-evident, Theorem 1 is useful because instead of trying to compute efficiency exactly we can look for a constant θ such that $T_{F_1}(D,n)/(T_F(D,n)) \leq \theta$, and $e_{F_1} \leq 1 - \theta$. As an application of the theorem, we can think of the CM itself as a filter F. F is very accurate (gets all the true hits) and efficient (random sequences do not score high), but slow ($T_F(D, n) = \Omega(|D|n^2)$) (Klein and Eddy, 2003; Zhang *et al*., 2005). On the other hand, the HMM filter F_1 is accurate ($A_F = A_{F_1}$), and an order of magnitude faster ($T_{F_1}(D,n) = O(|D|n)$), but not as efficient. Can the composite filter dominate? Note that $T_{F_1}(D,n)/(T_F(D,n)) \leq 1/n$. From Theorem 1, the composite filter $F_1 \cdot F$ dominates F if $e_{F_1} \leq (n-1)/n$. As this condition is relatively easy to achieve, Weinberg and Ruzzo show improvements for most families (Weinberg and Ruzzo, 2004a). In the following, we will describe sequence based filters that run in time $c|D|$, where c is a small constant. By the previous argument, we only need to show marginal efficiency $(n-c)/n$ to dominate. Thus, the filters we design will dominate the HMM filters of Weinberg and Ruzzo (2004a).

3 SEQUENCE FILTERS

Let F_P denote a sequence based filter, which computes a gapped SEQSCORE, and uses a threshold T, chosen so that the accuracy of F_P is identical to the CM. We will define a sequence based filter F_s that matches the accuracy of F_P, but is faster. The idea is based on an application of the pigeonhole principle, and the fact that text search using a *dictionary* of words is fast. For a sequence to score T against a profile of length L, each column must score T/L on the average. In fact, every sequence that scores T against the profile contains an l-mer w that scores Tl/L or better against the profile. F_S proceeds by computing all subsequences that match at least one keyword in T. We use the following procedure:

(1) Generate a set of keywords K, each of length l (for a fixed parameter l), by selecting all words that score Tl/L in an ungapped region of the profile. Label each such keyword w so that LABEL(w) is the profile position where it occurs.
(2) Search D for exact matches to keywords from K.
(3) For each position i that matches a keyword with label p, identify $D[i-p, \ldots, i-p+L]$ as a candidate sequence.
(4) Merge significantly overlapping candidate sequences.

By the pigeonhole principle, the accuracy of F_S is high ($A_{F_P} \subseteq A_{F_S}$). The filtering can be done in $O(|D|)$ time through the use of Aho-Corasick tries, or hashing, so the filter time is an order of magnitude faster. It remains to evaluate the efficiency of this filter. For any position i to be selected, either of the keywords in K must match at a specific position (given by their label) relative to i. Therefore, assuming a uniform distribution of words along the sequence, the efficiency of this filter is given by $(\frac{|K|}{4^l})$. By Theorem 1 , we only require $\frac{|K|}{4^l} < \frac{n-1}{n}$ for dominance, and can often find single keyword filters that suffice. In the following we improve upon this simple filter by considering multiple keywords.

3.1 Multiple keyword (chain) filtering

We define an (l, m, δ, K)-*chain filter* as follows: sequence $D[i, \ldots, i+L]$ is accepted by an (l, m, δ, K)-chain filter if m words $w_1, w_2, \ldots, w_m \in K$, each of length l match at positions $i+i_1, i+i_2, \ldots, i+i_m$, s.t. for all j, $i_j \geq i_{j-1} + l$ (i.e., words are ordered and non-overlapping) and $|i_j - \text{LABEL}(w_j)| \leq \delta$. For ungapped alignments, $\delta = 0$, but otherwise, δ must be chosen carefully to maximize accuracy. We have the following result:

THEOREM 2. *Consider an (l, m, δ, s_K)-chain filter. If s_K is the maximum number of keywords with an identical label in K then the efficiency on a uniform random database is given by*

$$e_F(l,m,\delta,s_K) = \binom{L - m(l-1)}{m} \left(\frac{2\delta s_K}{4^l}\right)^m. \qquad (1)$$

PROOF. Consider a random position i in the database D. By definition,

$$e_F(l,m,\delta,s_K) = Pr[D[i, \ldots, i+L] \text{ is accepted}].$$

Define a *configuration* w.r.t. a position i as an m-tuple $C(i) = (i_1, i_2, \ldots, i_m)$, such that $i \leq i_1 \leq i_2 \ldots \leq i_m \leq i + L$ and $i_j \geq i_{j-1} + l$ for all j. Then i is accepted by the filter if there exists a configuration $C(i)$ such that for all $i_j \in C$, $D[i_j, \ldots, i_j + l - 1] = w_j$ for some $w_j \in K$ with $|\text{LABEL}(w_j) - i_j| \leq \delta$. Thus, the probability for i_j to match up by chance is $\frac{2\delta s_K}{4^l}$. It follows that the efficiency of the (l, m, δ, K)-chain filter is $C_m(2\delta s_K/4^l)^m$, where C_m is the number of possible configurations. To compute this number, consider a binary string b with exactly m ones and $L - lm$ zeros. For $1 \leq j \leq m$, let b_j be the position of the j-th '1' from the left. Define $i_j = b_j + (j-1)l$. Then each binary string corresponds to a unique m-tuple $(i_1, i_2, \ldots, i_m)$, and $i_{j+1} - i_j = b_{j+1} + b_j + l \geq l$ for all $j < m$. The number of configurations is equal to the number of distinct binary strings, given by $C_m = \binom{L - m(l-1)}{m}$.

Figure 1 shows (as expected) that the efficiency of a chain filter F_C decreases exponentially with increasing m. The slightly faster than exponential decay is due to the fact that $L - ml$ also decreases with increasing m. Likewise, higher values of s_K decrease the rate of decay. However, for multiple keywords, selecting the set K of keywords becomes a challenging problem. The pigeonhole principle guarantees the existence of m words that score at least mTl/L, but does not bound the minimum score on any single word. If we were to choose K to be the set of all keywords, s_K could be prohibitively large. On the other hand, any choice of a lower bound will reduce Accuracy ($A_{F_P} \not\subseteq A_{F_C}$). In practice, there are many reasonable choices that ensure that the accuracy remains 1 and high efficiency is maintained. Currently, the features we deploy use empirically chosen cut-offs for keyword scores. However, there is a principled

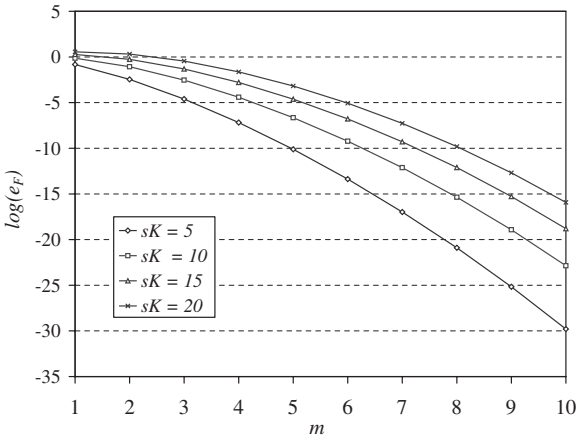


Fig. 1. A plot of $\log(e_F)$ versus m, when $L = 150$, $l = 8$ and $\delta = 20$. Different lines correspond to different values of s_K.

way to get around this obstacle by using an appropriately chosen *union* of filters.

3.2 Accuracy of chain filters

To control the accuracy of chain filters we extend their definition to allow a score threshold S, such that a sequence is accepted by the filter if in addition to satisfying the above conditions the total score of the matched keywords exceeds S. Let $\theta = Tl/L$. We are interested in computing a chain of words that score $m\theta$. We illustrate the approach using a parameter $\theta' = \theta/2$. Any subsequence that is accepted by the chain-filter must have some k ($1 \leq k \leq m$) words $w_1, \ldots, w_k$ that each score at least θ'. Let $w_{k+1}, \ldots, w_m$ denote the remaining words in the chain filter. We have

$$m\theta \leq \sum_{j=1}^{k} \text{SCORE}(w_j) + \sum_{j=k+1}^{m} \text{SCORE}(w_j)$$
$$\leq \sum_{j=1}^{k} \text{SCORE}(w_j) + (m-k)\theta'.$$

Thus $\sum_{j=1}^{k} \text{SCORE}(w_j) \geq m\theta - (m-k)\theta' = (m+k)\theta/2$.

For all $1 \leq k \leq m$, define an extended chain filter F_k of k words in which each word scores at least $\theta/2$, and the chain must score at least $(m+k)\theta/2$. Observe that $F_1 + \ldots + F_m$ accepts every chain that scores above $m\theta$, implying that $A_{F_C} \subseteq A_{F_1+\ldots+F_m}$. In the next section, we show that chain filters can be computed efficiently, in time that is often $o(|D|n)$. The search time of the union filter grows linearly with m, and so an efficiency/speed trade-off must be considered in selecting an appropriate m. Once again, Theorem 1 can be used to ensure dominance, but we must do it in an empirical setting as the running time depends upon the score distribution of keywords in K, which in turn, depends upon the alignment. Our results in Section 5 show that dominating filters are easy to find.

3.3 Implementing chain filters

We wish to filter substrings that match an extended (l, m, δ, K, S)-chain filter (where S is the score threshold). Our goal is to improve upon the profile search time of $O(L|D|)$. As chain filters are based on matches with l-mers, we can improve the speed by using string matching techniques. The algorithm is as follows:

(1) Build an Aho-Corasick Trie T_K with K (alternatively, if l is small, construct a hash table for occurrences of l-mers in D).

(2) Initialize a set of *active* intervals $\mathcal{I} = \phi$.

(3) Scan D with T_K. For each hit of word $w \in K$ at position i, add the interval $\pi = [i - \text{LABEL}(w) - \delta, i - \text{LABEL}(w) + \delta]$ to $\mathcal{I}$. The score of the interval $\text{SC}[\pi]$ is set to the score of w against the profile. Also, set the position as $\text{POS}[\pi] = i$.

(4) For each position $j \in D$, let $\mathcal{I}_j = \{\pi \mid j \in \pi\}$ be the subset of intervals that overlap with j. For most choices of parameters, $|\mathcal{I}_j| << L$. Select position j if there exist m intervals that are disjoint and have net score better than m. This is done as follows:

  (1) Sort the intervals in $\mathcal{I}_j$ according to $\text{POS}[\pi]$. For each $\pi \in \mathcal{I}_j$, let $p_1(\pi)$ be the largest interval with $\text{POS}[\pi] - \text{POS}[p_1(\pi)] > l$, and $p_2(\pi)$ be the predecessor of π.

  (2) for all intervals $\pi \in \mathcal{I}_j$ $\text{SCORE}[j, \pi] = \max\{\text{SC}[\pi] + \text{SCORE}[j, p_1(\pi)], \text{SCORE}[j, p_2(\pi)]\}$. Output j, if $\text{SCORE}[j, \pi]$ exceeds the threshold.

The entire computation takes time $\sum_j |\mathcal{I}_j| = o(L|D|)$. Also, the computation is done only if the depth of coverage at position j exceeds a threshold. The depth of coverage can be computed in linear time. This discussion hides an important problem. Insertions and deletions make the profile length significantly longer than any sequence. For example, the average length of cobalamin riboswitches is 200 , while the profile length is closer to 600 . A simple way around this is to discard columns with many gap characters, but that entails deciding which columns are dominated by gaps. Instead, we revise the definition of the LABEL of a position. Recall that LABEL of a keyword is its position in the profile, and should match its position in the query sequence. Instead, define the LABEL as the expected position in the query sequence. Let p_i denote the probability that the i-th position of the profile is not a gap (in other words, $p_i = P[i, A] + P[i, C] + P[i, G] + P[i, T]$). Then define

$$\text{label}_i = \begin{cases} p_i & \text{if } i = 1, \\ \text{label}_{i-1} + p_i & \text{otherwise.} \end{cases}$$

Each keyword that appears at position i in the profile is assigned label_i as its label.

4 RNA-PROFILE SCORING AND ALIGNMENT

In this section we describe our algorithm for scoring a sequence against a structural alignment of an RNA family, where we score for conservation of both sequence and structure. The algorithm is very similar to Covariance Model (Durbin *et al*., 1998; Eddy, 2002). However, we provide our own implementation to allow for faster banded scoring. Also, our filter design can be more effectively tied to the scoring. Formally, we treat the RNA-profile alignment as a filter, and compose it with the chain filter. Finally, our algorithm can be extended to include more complex RNA models, such as pseudoknots, which will be explored in future work.

The structural alignment of an RNA family is a (gapped) multiple alignment R of its sequences with structure described by a set M of pairs of positions (i, j), such that for a majority of sequences in the family, the nucleotides aligning to these positions form base-pairs.

procedure PAln

(*M is the set of base-pairs in RNA profile R. M' is the augmented set. *)

for all intervals (i, j), $1 \le i < j \le n$, all nodes $v \in M'$

if $v \in M$

$$A[i,j,v] = \max \begin{cases} A[i+1, j-1, \text{child}(v)] + \delta(l_v, r_v, t[i], t[j]), \\ A[i, j-1, v] + \gamma('-', t[j]), \\ A[i+1, j, v] + \gamma('-', t[i]), \\ A[i+1, j, \text{child}[v]] + \gamma(l_v, t[i]) + \gamma(r_v, '-'), \\ A[i, j-1, \text{child}[v]] + \gamma(l_v, '-') + \gamma(r_v, t[j]), \\ A[i, j, \text{child}[v]] + \gamma(l_v, '-') + \gamma(r_v, '-'), \end{cases}$$

else if $v \in M' - M$, and v has one child

$$A[i,j,v] = \max \begin{cases} A[i, j-1, \text{child}[v]] + \gamma(r_v, t[j]), \\ A[i, j, \text{child}[v]] + \gamma(r_v, '-'), \\ A[i, j-1, v] + \gamma('-', t[j]), \\ A[i+1, j, v] + \gamma('-', t[i]), \end{cases}$$

else if $v \in M' - M$, and v has two children

$$A[i,j,v] = \max_{i \le k \le j} \{A[i, k-1, \text{left\_child}[v]] + A[k, j, \text{right\_child}[v]]\}$$

end if

end for

Fig. 2. An algorithm for aligning an RNA profile R with m columns against a database string t of length n. The query consensus structure M has been *Binarized* to get M'. Each node v in the tree corresponds to a base-pair $(l_v, r_v) \in M'$.

The alignment of the RNA family against a target sequence t is described by a $2 \times m$ matrix A, in which row 1 contains *column positions* of the profile interspersed with spaces (insertion of aligned sequence), and row 2 contains the *sequence*, also interspersed with spaces (deletion of profile columns). For all columns j, $A[1, j] \neq' -'$ or $A[2, j] \neq' -'$. For $r \in \{1, 2\}$, define $\rho_r[j] = j - |\{l < i \text{ s.t. } A[r, l] =' -'\}|$. In other words, if $A[1, j] \neq' -'$, it contains the position $\rho_1[i]$ of R. The score of alignment A is given by

$$\sum_j \gamma(A[1,j], A[2,j]) + \sum_{(\rho_1[i], \rho_1[j]) \in M} \delta(\rho_1[i], \rho_1[j], \rho_2[i], \rho_2[j]).$$

The function γ scores for sequence similarity, and δ scores for structure conservation. Our goal is to find an alignment that maximizes this score. While this formulation encodes a linear gap penalty, we note here that alignments of RNA molecules may contain large gaps, particularly in the loop regions, and we implement affine penalties for gaps (details omitted).

4.1 Choosing the scoring functions

Consider an alignment of n RNA sequences from a family. Let $n_i(a)$ be the number of sequences with $a \in \{A, C, G, U,' -'\}$ in the i-th column of the multiple alignment. The probability of observing a in the i-th position can be estimated by

$$P_i(a) = \frac{C_a + n_i(a)}{\sum_{a'} C_{a'} + n}$$

where C_a are pseudo-counts, chosen so that $p_a = C_a / (\Sigma_{a'} C_{a'})$, where p_a is the probability of occurrence of a in the family. These probabilities are used to construct a position specific scoring matrix. Then for all positions i, and al symbols $a \in \{A, C, G, U,' -'\}$

$$\gamma(i, a) = \sum_{a' \in \{A, C, G, T, -\}} S(a', a) \times P_j(a') \qquad (2)$$

where $S(a', a)$ is the score of substituting a' with a. We use a nucleotide substitution scoring matrix (Klein and Eddy, 2003). We model insertions and deletions with the gap penalties $\gamma('-', a)$, and $\gamma(i,'-')$, respectively.

Likewise, to score for structure conservation we look at the probabilities of specific base-pairs that occur in each pair of positions. For each $(i, j) \in M$, let $n_{i,j}(a, b)$ describe the number of sequences in the alignment that contain a in position i, and b in position j. As before,

$$P_{i,j}(a, b) = \frac{C_{a,b} + n_{i,j}(a, b)}{\sum_{a', b' \in \{A, C, G, U, '-'\}} C_{a', b'} + n}$$

and the score for conserved structure is given by

$$\delta(i, j, a, b) = \sum_{a', b' \in \{A, C, G, U\}} P_{i,j}(a', b') \times S_p(a', b', a, b) \qquad (3)$$
$$\forall (i, j) \in M, a, b \in \{A, C, G, U\}$$

where S_p is scoring matrix for substituting (a', b') with (a, b), and rewards both sequence and structure conservation. Note that δ is only defined when $(i, j) \in M$, and $a, b \in \{A, C, G, U\}$. In other cases, the structure is obviously not conserved, and the appropriate score is given by γ.

4.2 The alignment procedure

We make the assumption that the base-pairs are non-crossing. For each base-pair $(i, j) \in M$, there is a unique (*parent*) base-pair (i', j') such that $i' < i < j < j'$, and there is no base-pair (i'', j'') such that $i < i'' < i'$, or $j < j'' < j'$. Thus the alignment can be done by recursing on the nodes of the tree. However, the tree can have high degree and not all columns of the profile participate in it. To this end we binarize the tree using the procedure given in Zhang *et al.* (2005). Specifically, we add spurious nodes (base-pairs) to the tree so that every column participates as a tree node, the degree of any node is at most 3 , and the number of nodes is $O(m)$, where m is the number of columns in the profile. Further, a node corresponding to a true base-pair $(i, j) \in M$ has at most one child.

Table 1. Riboswitch sub-families in the Rfam database (version 7.0)

Rfam Id	Name	Average length	%id	#seed	#total
RF00050	FMN	145	66	48	136
RF00059	TPP	110	52	237	382
RF00080	yybP-ykoY	128	45	74	127
RF00162	SAM	110	67	71	219
RF00167	Purine	100	56	37	100
RF00168	Lysine	182	49	60	98
RF00174	Cobalamin	204	46	171	249
RF00234	glmS	184	58	14	37
RF00379	ydaO-yuaA	158	54	35	74
RF00380	ykoK	168	60	39	53
RF00442	ykkC-yxkD	106	62	16	21
RF00504	gcvT	101	51	117	163

Average length and ''%identity'' are based on the information in the Rfam database. '#seed' is the number of sequences in the seed alignment. '#total' is the number of full family sequences.

Figure 2 describes a dynamic programming algorithm for aligning a sequence to an RNA profile. The RNA profile is described by a tree. Each node v in the tree either corresponds to a base-pair $(l_v, r_v) \in M'$ of the profile, where M' is the augmented list of base-pairs. The alignment of the sequence to the RNA profile is done by recursing on the tree like structure of RNA. Each node in the binarized tree either represents a base-pair/unpaired base (and has its own PSSM), or represents a branching point in a pair of parallel loops. The algorithm maintains the sequence interval being aligned and the current node in the structure tree.

5 EXPERIMENTAL RESULTS

We implemented the chain filtering and the profile alignment algorithms as described above. All tests reported herein were performed on a 2.8 GHz Intel PC (genomic searches were done on 1.6GHz AMD Opteron grid). For chain filtering, we chose the parameters l, m, δ and score threshold (affects s_K) so as to optimize efficiency while maintaining optimal accuracy. The chain filtering was also composed with HMM filtering (from RAVENNA package (Weinberg and Ruzzo, 2004a)) to further improve the filtering efficiency. For the alignment of the filtered sequences to an RNA model we used both our profile alignment tool and the CMsearch tool from the INFERNAL suite (http://infernal.wustl.edu) Eddy (2002); Griffiths-Jones *et al*. (2003). Both the HMM filters (using expended HMM filters) and CMsearch were applied in the following with their default parameters or recommended parameters from the Rfam database website.

We applied these algorithms to search for riboswitch elements. We chose to focus on riboswitches both due to their importance and due to their unique properties that make them an ideal test case: many ncRNA families show strong sequence similarity, which makes sequence based filtering very efficient, and relatively trivial. In contrast, the riboswitches, with 12 distinct sub-families (and new sub-families being continuously discovered) are quite diverse, and relatively difficult to filter. Table 1 summarizes known riboswitches from the Rfam database, version 7.0 (Griffiths-Jones *et al*., 2003, 2005).

5.1 Filter efficiency and accuracy

To systematically test our filters, we downloaded data on 12 riboswitch sub-families from the Rfam database, version 7.0 (Griffiths-Jones *et al*., 2003, 2005). These data contain for each family a 'seed' alignment, which is a hand-curated alignment of known members, and a 'full' collection of family sequences, which contains known and predicted (by CMsearch) members. In the following we refer to a member of the seed alignment as *seed sequence*, and to a member of the full collection as *family sequence*.

Synthetic databases: As a first test of our method we synthesized several test sequences. For each sub-family, we created a random genomic sequence of size 1 Mb with G+C content of 0.5, and randomly planted all the family sequences therein. We tested the filter's performance on the composite sequence. Table 2 summarizes the results of the chain filter (CF) in comparison to the HMM filters and to a combined filter. In addition to the efficiency measure we also report a second measure *efficiency*2, which is computed exclusively on the random sequence. While the actual genomic sequence will have some true hits as well, it is unlikely to have more than a few members per Mb, so efficiency2 is a better approximation to the true efficiency.

Recall from Theorem 1 that high gains in filter speed at the cost of efficiency is desirable because filter composition can be used to achieve dominance. Thus, the key statistic in Table 2 is search time. The sequence based chain filter is much faster (on average, 9 sec/Mb) than the HMM filter (71 sec/Mb). Interestingly, even the efficiency of CF filter remains very high on the average (0.036) while maintaining optimal accuracy. The faster speed and the optimal accuracy of the CF filter makes the composite filter (CF·HMM), which applies CF filter first and HMM filter later on the database, dominate the HMM filter. In Table 2, CF·HMM further improves the efficiency significantly (0.029), and it is still much faster (on average, 14 sec/Mb) than the HMM filter. The filtering is followed by alignment with RNA-Profile. We also include a direct comparison between profile alignment and the CM approach. As can be seen from Table 3, profile alignment attains very similar accuracies but is much faster.

Genomic sequences: Next, we tested the performance of our filter on two genomes with biased G+C content, previously used by Weinberg and Ruzzo (2004)a: *E. coli* K12 and *Staphylococcus aureus* MW2. We searched for the 12 riboswitch families on these genomes whose total length is 7.5 Mb. Table 4 presents a comparison to the HMM filter. As expected, the chain filter is much faster. On the average, its efficiency is also very high (0.017), outperforming that of the HMM filter (0.34). Note that all true hits in these two genomes were recovered by every filtering method with the corresponding alignment algorithm. Obviously, the composite filter, CF·HMM, still provides the fastest filtering solution.

5.2 Discovering novel riboswitches

We applied our sequence based filters, coupled with profile alignment, to search all bacterial and archaeal genomes for the twelve riboswitch families. A total of 254 genomes spanning 818 Mb were searched. Of these, 179 have some ncRNA annotations. Table 5 summarizes the search results. In total we identified 463 novel (putative) riboswitches based on a P-value cutoff 0.04. Interestingly, 413 of these predictions were within 500 bp upstream of an annotated gene. These predictions include hits to

Table 2. Filtering performance of chain filters (CF), HMM filters (HMM), and composite filters (CF·HMM) on synthetic sequences

Family	CF				HMM				CF·HMM		
	eff.	eff2.	acc	time(m:s)	eff.	eff2.	acc.	time(m:s)	eff.	eff2.	time(m:s)
FMN	1.3e-2	0	1	0:10	2.8e-2	0	1	1:10	1.3e-2	0	0:11
TPP	6.3e-2	3.4e-2	1	0:07	1	1	1	0:59	5.8e-2	3.1e-2	0:14
yybP-ykoY	1.5e-1	1.4e-1	1	0:08	1	1	1	1:07	1.4e-1	1.3e-1	0:28
SAM	1.8e-2	2.1e-3	1	0:07	5.9e-2	4.0e-4	1	0:55	1.7e-2	0	0:09
Purine	3.8e-2	3.1e-2	0.99∗	0:7	1.1e-2	1.5e-4	1	0:52	7.4e-3	5.9e-5	0:10
Lysine	1.5e-2	3.9e-3	0.99∗	0:10	1	1	1	1:34	1.5e-2	3.8e-3	0:13
Cobalamin	6.3e-2	3.4e-2	1	0:13	1	1	1	1:42	6.2e-2	3.3e-2	0:26
glmS	1.3e-2	9.1e-3	1	0:14	7.7e-3	3.0e-4	0.97	1:25	2.4e-3	0	0:17
ydaO-yuaA	1.2e-2	4.9e-3	1	0:08	1.9e-2	1.0e-3	1	1:11	6.9e-3	7.5e-5	0:10
ykoK	1.2e-2	6.0e-3	1	0:10	1.2e-2	1.2e-4	1	1:32	5.9e-3	0	0:12
ykkC-yxkD	1.7e-3	0	1	0:07	2.4e-3	0	1	0:53	1.7e-3	0	0:07
gcvT	3.7e-2	2.5e-2	1	0:07	1.9e-1	1.6e-1	1	0:51	1.6e-2	4.3e-3	0:10
Average	**0.036**	**0.024**	**1**	**0:09**	**0.361**	**0.347**	**1**	**1:11**	**0.029**	**0.017**	**0:14**

'eff.' is the efficiency on synthetic sequences, 'eff2.' is the efficiency on exclusively random sequences, 'acc.' is the accuracy on synthetic sequences, and 'time' is the running time on synthetic sequences. (∗) Note that these filters only miss one hit.

Table 3. Comparison of RNA profile alignment (PAln) and CMsearch (CM) on synthetic sequences

Family	PAln #TP	PAln #true	PAln retrieval rate	CF· PAln time (m:s)	CM #TP	CM #true	CM retrieval rate	HMM· CM time (h:m:s)
FMN	136	136	1	1:29	136	136	1	13:24
TPP	373	382	0.98	6:06	382	382	1	7:06:47
yybP-ykoY	119	127	0.94	14:43	127	127	1	4:11:31
SAM	218	219	1	2:23	219	219	1	12:03
Purine	99	99	1	2:17	100	100	1	2:05
Lysine	97	98	0.99	3:16	98	98	1	13:57:59
Cobalamin	242	249	0.97	14:58	248	249	1	27:39:27
glmS	36	37	0.97	2:36	35	37	0.95	6:53
ydaO-yuaA	73	74	0.99	3:15	73	74	0.99	13:16
ykoK	52	53	0.98	1:22	53	53	1	8:39
ykkC-yxkD	21	21	1	0:30	21	21	1	1:56
gcvT	138	163	0.85	3:15	163	163	1	37:48

RNA profile alignment uses p-value cut-off 0:05 to get the top ranking hits (one hit in cobalamin family is marginal), and CMsearch use the same cutoff bits score from Rfam data website. 'retrieval rate' is defined as the percentage of true positive (#TP) hits over the possible true hits (#true) after filtering (either chain filtering (CF) or HMM filtering (HMM)).

genomes that had previously been annotated for ncRNA in Rfam. For cobalamin riboswitch (as an example), most of the predictions are, indeed, in 5' UTRs of cobalamin-related or cobalamin-associated genes (Rodionov *et al*., 2003b; Vitreschak *et al*., 2003) (B12 synthesis, cobalt transporters and alternative cobalamin-independent enzymes). One of the predicted cobalamin riboswitches has been experimentally tested and confirmed (data not shown). In the gcvT (glycine-dependent riboswitch) family, we found 28 novel hits, of which 12 occur as proximal pairs, which is known a preferred mechanism of action for this family (Mandal *et al*., 2004). Detailed information on these discoveries is presented in supplementary data (http://www.cse.ucsd.edu/~shzhang/paper/ISMB2006).

6 CONCLUSIONS

We reiterate that the main contribution of this paper is not simply to provide improved filtering, but to formalize the filtering problem, and demonstrate that a simple approach based on combining filters is useful. While our results improve the state-of-the-art and are likely to be useful in discovering novel ncRNAs, many questions remain unanswered. Some of the open problems are directly related to our analysis. First, can we give theoretical bounds on the efficiency vs. speed trade-off for the union filters? This will probably entail some assumptions on the distribution of keyword scores. Second, can we design optimal chain filters, which provably dominate all other sequence based filters? Indeed the bulk of the results presented here are presented on filters that are fast, but not perhaps as efficient as could be. On the other hand, HMMs are efficient, but not always fast, which indicates that there is room for more filters in between. Examples of such filters include subsets of profiles (choose a subset of contiguous conserved columns, and filter based on those), or a hierarchy of compositions instead of a single one. Finally, for the most diverse families, it is likely that sequence based filters will not be efficient. Fast filters based on structure considerations have been shown to be effective (Weinberg and

Table 4. Filtering performance of chain filters (CF), HMM filters (HMM) and composite filters (CF·HMM) on two real genomes with alignment performance of profile alignment (PAln) and CMsearch (CM)

Family	CF eff.	CF time (m:s)	'CF· PAln' time (m:s)	HMM eff.	HMM time (m:s)	'HMM· CM' time (m:s)	'CF· HMM· CM' time (m:s)	'CF· HMM· PAln' time (m:s)	CM· time estimated (hours)
FMN	1.2e-4	1:28	2:24	9.1e-6	15:40	16:10	1:32	2:24	54h
TPP	2.5e-2	1:10	10:39	9.9e-1	12:03	45h	16:10	7:55	40h
yybP-ykoY	6.7e-2	1:20	42:23	1	13:55	44h	53:04	36:59	38h
SAM	4.5e-4	1:11	2:01	8.2e-4	11:34	12:39	1:15	1:51	31h
Purine	2.8e-2	1:08	10:38	2.7e-3	10:51	12:36	1:34	1:49	18h
Lysine	3.8e-3	1:41	6:06	1	19:23	100h	6:28	5:46	85h
Cobalamin	4.0e-2	1:55	44:56	9.6e-1	20:31	177h	69:42	39:40	166h
glmS	1.9e-2	2:00	13:09	2.1e-3	17:53	18:28	2:24	3:32	80h
ydaO-yuaA	3.8e-3	1:20	6:27	3.5e-4	11:00	11:47	1:22	3:10	99h
ykoK	3.0e-3	1:21	5:32	1.0e-3	114:00	16:19	1:26	3:00	61h
ykkC-yxkD	0	1:00	1:29	0	8:19	8:23	1:00	1:29	23h
gcvT	1.5e-2	0:54	6:49	1.5e-1	7:45	2.4h	2:01	2:37	33h
Average	**0.017**	**1:22**	**12:42**	**0.34**	**14:05**	**31h**	**13:10**	**9:11**	**61h**

'eff.' is the filtering efficiency, and 'time' is the running time of the corresponding filters. Note that when HMM filter efficiency is close to 1, the computation time for HMM · CM (from RAVENNA (Weinberg and Ruzzo, 2004*a*) package) is longer than the computation time for CMsearch (from Infernal package (Griffiths-Jones *et al*., 2003)). This is because of the differences between the C++ and the C compiler.

Table 5. Summary of searching riboswitches against the whole bacterial and archaeal genomes

Family	#known	#TP	#new	#new*	CF eff.	CF·PAln time (hours)	CM time estimated (days)
FMN	103	92	34	2	8.5e-4	4.8	236.9
TPP	305	235	89	6	7.9e-3	6.7	232.4
yybP-ykoY	109	74	65	25	7.7e-2	63.7	166.5
SAM	204	182	80	3	6.7e-4	3.4	136.0
Purine	82	72	31	10	5.7e-2	34.3	82.6
Lysine	82	61	23	5	5.7e-3	12.6	405.8
Cobalamin	189	141	70	15	3.6e-2	65.1	794.0
glmS	24	23	8	1	1.4e-3	6.9	372.1
ydaO-yuaA	68	62	17	57	2.3e-2	36.9	470.2
ykoK	44	39	7	2	3.9e-3	10.5	266.7
ykkC-yxkD	14	14	11	1	1.4e-5	2.8	98.7
gcvT	148	98	28	1	4.2e-2	27.2	136.8

'#known' is the number of known riboswitches in the whole bacterial and archaeal genomes, '#TP' is the number of predicted known hits, '#new' is the number of new predictions in these genomes, and '#new∗' is the number of new predictions from the genomes that had previously been annotated for ncRNA in Rfam.

Ruzzo, 2004b; Zhang *et al*., 2005), but have been completely ignored in the present study. It is an important open problem to formalize their efficiency and speed, and to study their combination with sequence based filters. We hope that these and related challenges will spur the development of filters, and ultimately lead to better tools for mining biomolecular databases.

ACKNOWLEDGEMENTS

This work is supported by a grant from the National Science Foundation (NSF-DBI:0516440) (S.Z. and V.B) and by an Alon Fellowship (R.S.). This research is also supported in part by the UCSD FWGrid Project (NSF Research Infrastructure Grant Number EIA-0303622).

REFERENCES

Bafna,V., Tang,H. and Zhang,S. (2006) Consensus folding of unaligned RNA sequences revisited. *J. Comput. Biol.*, **13** (2), 283–295.

Dost,B., Han,B., Zhang,S. and Bafna,V. (2006) Structural alignment of pseudoknotted RNA. In *RECOMB '06: Proceedings of the eighth annual international conference on Resaerch in computational molecular biology,* Springer New York, NY, USA, 143–158.

Durbin,R., Eddy,S.R., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis*. Cambridge University Press.

Eddy,S.R. (2001) Non-coding RNA genes and the modern RNA world. *Nat. Rev. Genet*., **2**, 919–929.

Eddy,S.R. (2002) A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. *BMC Bioinformatics*, **3**, 18.

Frank,A., Tanner,S., Bafna,V. and Pevzner,P. (2005) Peptide sequence tags for fast database search in mass-spectrometry. *J. Proteome Res*., **4** (4), 1287–1295.

Griffiths-Jones,S., Bateman,A., Marshall,M., Khanna,A. and Eddy,S.R. (2003) Rfam: an RNA family database. *Nucleic Acids Res*., **31**(1), 439–441.

Griffiths-Jones,S., Moxon,S., Marshall,M., Khanna,A., Eddy,S.R. and Bateman,A. (2005) Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res*., **33** (Database issue), 121–124.

Klein,R.J. and Eddy,S.R. (2003) Rsearch: finding homologs of single structured rna sequences. *BMC Bioinformatics*, **4** (1), 44.

Leibowitz,N., Fligelman,Z.Y., Nussinov,R. and Wolfson,H.J. (1999) Multiple structural alignment and core detection by geometric hashing. In *Proc. Int. Conf. Intell. Syst. Mol. Biol*. pp. 169–177.

Mandal,M., Lee,M., Barrick,J.E., Weinberg,Z., Emilsson,G.M., Ruzzo,W.L. and Breaker,R.R. (2004) A glycine-dependent riboswitch that uses cooperative binding to control gene expression. *Science*, **306** (5694), 275–279.

Nahvi,A., Sudarshan,N., Ebert,M., Zou,X., Brown,K. and Breaker,R. (2003) Genetic control by a metabolite binding mRNA. *Chem. Biol*., **9**, 1043–1049.

Pedersen,J.S., Bejerano,G., Siepel,A., Rosenbloom,K., Lindblad-Toh,K., Lander,E.S., Kent,J., Miller,W. and Haussler,D. (2006) Identification and Classification of Conserved RNA Secondary Structures in the Human Genome. *PLoS Comput. Biol*., **2**(4), e33.

Rivas,E. and Eddy,S. (2000) Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics*, **16**(7), 583–605.

Rivas,E. and Eddy,S.R. (2001) Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, **2**, 8.

Rodionov,D.A., Vitreschak,A.G., Mironov,A.A. and Gelfand,M.S. (2003a) Regulation of lysine biosynthesis and transport genes in bacteria: yet another RNA riboswitch? *Nucleic Acids Res*., **31** (23), 6748–6757.

Rodionov,D.A., Vitreschak,A.G., Mironov,A.A. and Gelfand,M.S. (2003b) Comparative genomics of the vitamin B12 metabolism and regulation in prokaryotes. *J. Biol. Chem*., **278** (42), 41148–41159.

Storz,G. (2002) An expanding universe of noncoding RNAs. *Science*, **296** (5571), 1260–1263.

Sudarsan,N., Wickiser,J.K., Nakamura,S., Ebert,M.S. and Breaker,R.R. (2003) An mRNA structure in bacteria that controls gene expression by binding lysine. *Genes Dev*., **17** (21), 2688–2697.

Tanner,S., Shu,H., Frank,A., Wang,L.C., Zandi,E., Mumby,M., Pevzner,P.A. and Bafna,V. (2005) InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem*., **77** (14), 4626–4639.

Vitreschak,A.G., Rodionov,D.A., Mironov,A.A. and Gelfand,M.S. (2003) Regulation of the vitamin B12 metabolism and transport in bacteria by a conserved RNA structural element. *RNA*, **9** (9), 1084–1097.

Vitreschak,A.G., Rodionov,D.A., Mironov,A.A. and Gelfand,M.S. (2004) Riboswitches: the oldest mechanism for the regulation of gene expression? *Trends Genet*., **20** (1), 44–50.

Washietl,S., Hofacker,I.L., Lukasser,M., Huttenhofer,A. and Stadler,P.F. (2005) Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome. *Nat. Biotechnol*., **23** (11), 1383–1390.

Weinberg,Z. and Ruzzo,W.L. (2004a) Faster genome annotation of non-coding rna families without loss of accuracy. In: *RECOMB '04: Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, ACM Press New York, NY, USA, pp. 243–251.

Weinberg,Z. and Ruzzo,W.L. (2004b) Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. *Bioinformatics*, **20** (Suppl 1), I334–I341.

Zhang,S., Hass,B., Eskin,E. and Bafna,V. (2005) Searching genomes for non-coding rna using fastr. *IEEE/ACM Trans. on Comput. Biol. and Bioinformatics*, **2** (4), 366–379.

AUTHOR INDEX

Vol. 22 no. 14 2006, pages e567–e568
doi:10.1093/Bioinformatics/btl375

