Poster H-51

Constrained Alignment Using Cluster Tree Elimination

Authors:
Sebastian Will (*University of Freiburg*)
Anke Busch (*University of Freiburg*)
Rolf Backofen (*University of Freiburg*)

Short Abstract: We revisit sequence comparison with side constraints. Often, when prior knowledge about the molecules is available, it is desirable to incorporate this information into the alignment. We introduce a constraint-based approach that allows for easy integration of various kinds of additional information, while preserving the efficiency of dynamic programming.

Long Abstract:

The comparison of macromolecules by sequence alignment is one of the most commonly used computational techniques in molecular biology. Regarding the standard case of sequence alignment, the problem is well understood and efficiently solved by dynamic programming (DP). However, in many cases prior knowledge about the molecules is available and one would like to take it into account. In particular, since automatic alignment methods can not produce biological meaningful alignments in all cases, additional knowledge helps to improve alignment quality. For example, the additional knowledge may be due to an expert biologists experience or due to reasoning over sequence annotations. More concretely, we could already know that certain segments of the sequences match very well to each other or improve RNA comparison by employing knowledge on the molecule structure [Sankoff, SIAM J.Appl.Math., 1985; Jiang et al., JCB, 2002; Backofen, Will, JBCB, 2004; Lancia et al., RECOMB, 2001]. Several groups discussed the incorporation of specific side constraints into alignment algorithms, i.e. constrained alignment. For example, Morgenstern et al. [Morgenstern et al., Bioinformatics, 2004] discussed anchor constraints and Myers et al. [Myers, et al., RECOMB, 1997] investigated alignment with precedence constraints. Anchor constraints tell that certain two residues can only match to each other and everything left (resp. right) of the first residue can only match with something left (resp. right) of the second residue. Precedence constraints tell that in the alignment a certain residue of the first sequence occurs left of a certain residue in the second sequence. Our approach to constrained alignment goes far beyond this earlier work, since almost arbitrary constraints can be incorporated. We achieve the generality of our approach by a declarative formulation of the problem. Often, declarative, in particular constraint-based, programming can offer improved extensibility over more traditional approaches. Commonly, the non-declarative method DP is applied to aligning sequences and sequences with additional information [Needleman, Wunsch, JMB, 1970; Jiang et al., JCB, 2002]. However, there is no straightforward and general way to extend a DP algorithm in order to take additional knowledge into account. Declarative formulations of the alignment problem have been proposed, in order to overcome the inflexibility of DP. Such approaches can be extended to incorporate prior knowledge in the form of constraints. One such previous approach is based on integer linear programming (ILP). Since ILP only uses boolean variables, the ILP model of

[Lenhof, Reinert, Vingron, RECOMB, 1998] for aligning two sequences of length n and m introduces O(nm) variables for modeling the alignment edges. Due to the resulting complexity, one needs to introduce artificial restrictions on the possible alignment edges for solving the problem in practice. Furthermore, the solving strategy for ILP does not achieve the efficiency of DP for the unconstrained case. Another declarative approach [Yap, Constraints, 2001] is based on constraint programming. The approach introduces quadratically many variables and constraints and remodels the given DP algorithm. As a consequence, only a rather restricted class of side constraints can be handled efficiently. Our constraint-based, declarative formulation of alignment overcomes both, the inflexibility of dynamic programming and the efficiency problems and limitations in expressivity of earlier declarative approaches. We model the problem of aligning two sequences a of length n and b of length m as a semi-ring based constraint problem. Therefore, we introduce variables X1, ..., Xn. Each variable has a value in the range [0..m]. For technical reason, we introduce auxiliary variables X0=0 and X(n+1)=m+1. A valuation of the variables represents an alignment of the two sequence, if and only if the values of the variables are in (non-strictly) ascending order. Then, position i in sequence a is aligned to position j in sequence b if and only if Xi=j and X(i-1)<=Xi holds and negative infinity otherwise. Furthermore, we introduce functions fi (1<=i<=n+1) on the variables X(i-1) and Xi that each yield a score contribution. Due to this encoding, the optimal alignment of a and b is represented by the valuation of our variables that maximizes the sum over all the functions fi and li (1<=i<=n+1). Next, we need an efficient and general solving strategy for such constraint models. This is provided by Cluster Tree Elimination (CTE). CTE can be applied to the basic model as well as to many possible extensions and each time yields a polynomial algorithm. We can even show that CTE can solve the basic model with O(nm) time and space complexity with only minor modifications. CTE utilizes a cluster tree decomposition (CTD) of the constraint model and works by sending messages along the tree. The CTD is basically a dependency tree that clusters variables and functions. We can give a simple, linear CTD for the basic model and derive the CTDs for extended models in a systematic way. The anchor and precedence constraints appear as rather simple extensions in our setting and are subsumed by our approach. The addition of both kinds of constraints does neither add variables nor change the CTD. Going beyond, we investigated segment constraints and structural constraints. The segment constraint tells that a segment of sequence a of length k matches a segment of sequence b by at least x%. Its addition to the constraint model requires to add k variables and k functions for counting matches. The incorporation of structural constraints introduces non-linear dependencies and therefore requires changes in the CTD. Besides its theoretically good complexity, the approach showed excellent performance in our prototypical implementation. Its main advantage over previous constrained alignment methods, is the handling of virtually arbitrarily complex new constraints and the free combination of different constraints in one alignment, while maintaining efficiency. We consider these features as essential for the flexible use of prior knowledge in sequence alignment.