Poster I-97

Fast Protein Structure Alignment by Joining Similar Substructure Pair

Authors:
Chan-Yong Park (*ETRI*)
Sung-Hee Park (*ETRI*)
Dae-Hee Kim (*ETRI*)
Soo-Jun Park (*ETRI*)
Hong-Ro Lee (*Chung Nam University*)
Chi-Jung Hwang (*Chung Nam University*)

Short Abstract: We propose a new fast protein structure alignment algorithms. This algorithm uses a 3D chain code representation for fast measuring the local geometric similarity of protein and apply a backtracking algorithm for joining a similar local substructure efficiently.

Long Abstract:
1. The Proposed Protein Structure Alignment Algorithm
The algorithm comprises four steps.
1.1 3D Chain Code
The protein structure data are obtained from the Protein Data Bank [4]. For each residue of the protein we obtain the 3D coordinates of its Cα atoms from the PDB file. As a result, each protein is represented by approximately equidistant sampling points in 3D space. To make a 3D chain code, we regard four Cα atoms as a set (Fig 2)[18]. We calculate a homogeneous coordinate transform to create a new coordinate (u,v,n) composed of a Up Vector (ci-ci+1) and a directional vector (ci+1, ci+2) as the new axis coordinate. For protein structure matching, the Ca atoms along the backbone can be considered as equally spaced because of the consistency in chemical bond formation. Since we can use the same polygonal length between the Ca atoms, we regard r as 1 in the spherical coordinate. By following this step, the 3D chain code (CCA) of protein A is created for a protein chain: Where n is the total number of amino acid of protein A minus 3.
1.2 Finding Similar Substructure Pair Set
Because the 3D chain code represents a relative direction in the 4 atoms of a protein, we can compare local similarity of two proteins by means of comparing 3D chain code of the two proteins. Given two proteins, we construct a similarity map. The similarity map represents how much two proteins are aligned together. The entry D (i,j) of the similarity map denotes the similarity between the 3D chain code values of the ith residue of protein A({Øi,θi}) and the jth residue of protein B({Øj,θj}), and is defined by the following equation. This measure is basically the Euclidian distance. After calculating each D(i,j) for for i and j, we obtain the entry value below than degree angles of threshold (Td) in similarity map. We use 10 as Td in our experiments. Figure 3 shows an example of a similarity map for the 3D chain code between two particular proteins called 1HCL and 1JSU:A. By using this similarity map, our goal is to find all SSPs in the map. A SSP is represented as a diagonal line in the map. For finding a SSP, we find first element D(i,j) with the value below Td and then, find the next element at D(i+1, j+1) and D(i-1, j-1) with the value below Td and the same procedure is

repeated until the next elements is below Td. This process can be viewed as finding diagonal lines in the similarity map. After finding a SSP, we define it as a SSPk I(i,j)

1.3 Merging Similar Substructure Pairs

In the previous section, we have found many SSPs. Because the computation time of the protein alignment depends on the number of SSPs, we merge specific SSPs into a SSP In the similarity map, we find rectangular shape which is composed of many SSPs. The SSPs which has same secondary structure cause the rectangular shape.(Figure 5) For example, if protein A and protein B has same α-helix structure, they have a similar geometric structure each 1 rotation turn. The β-strands are same. In this case, we merge SSPs with same secondary structure into a single SSP. After merging SSPs, the similarity map is shown in Figure 6.

1.4 Joining Similar Substructure Pairs

In this section, we should find optimal SSPs, which describe a possible alignment of protein A with protein B. We apply the modified backtracking algorithm [15] for joining SSPs. We represent the SSPs as nodes (vi) of the state space tree. The simple pseudo code is shown in figure 7. We use a connectivity value for each SSP as a promising function. If two SSPs (SSPk and SSPk+1) have a similar 3D rotation and translation below the threshold, the promising function returns the value true. The pseudo code is shown in figure 8. The root node in the tree is the first SSP. After running this algorithm, many solutions are established. We calculate the RMSD value from the solutions offered. Then, we select a solution with the minimum RMSD value.

2. Implementation and Results

Our empirical study of the protein structure alignment system using the 3D chain code could lead to very encouraging results. Figure 10 shows the alignment result of protein 1HCL_ and 1JSU_A. These protein are cyclin-dependent protein kinases, the uncomplexed monomer (1HCL:_) in the open state and the complex with cyclin and P27 (1JSU:A) in the closed state. While the sequences of the uncomplexed and complexed state are almost identical with 96.2% homology, there are significant conformational differences. Differences are found in both active site. The RMSD of the two proteins is 2.34 and the alignment time is 0.35 sec.

3. Discussion and Conclusion

This paper proposed a noble protein structure alignment method through the 3D chain code of a protein chain direction vector and a backtracking algorithm for joining SSPs. The 3D chain code represents the protein chain structure efficiently. The essential concept here is the idea of a protein chain as a thread. Beginning with this idea, we made a 3D chain code for searching similar substructures. For joining SSPs, we use the backtracking algorithm. Other protein structure alignment systems use dynamic programming. However, in this case, the backtracking algorithm is more intuitive and operates more efficiently. This algorithm has particular merit, unlike other algorithms. The methodology uses a 3D chain code that is more intuitive and a backtracking algorithm that is faster than dynamic programming generally speaking. Thus, the alignment is very faster. In general cases, the alignment time is 0.5 of a second and rarely exceeds 1.0 second.