Academic Sciences International Journal of Pharmacy and Pharmaceutical Sciences

ISSN- 0975-1491 Vol 3, Issue 4, 2011

Research Article

PROTEIN STRUCTURE ALIGNMENT USING DYNAMIC PROGRAMMING

NASRIN AKTER[1], MD. JAVED HOSSAIN[1]* AND MOHAMMAD SALIM HOSSAIN[2]

[1]Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali-3802, Bangladesh, [2]Department of Pharmacy, Noakhali Science and Technology University, Noakhali-3802, Bangladesh.
Email: javed_abc@yahoo.com

Received: 22 May 2011, Revised and Accepted: 4 July 2011

ABSTRACT

The three dimensional structure of protein can yield direct insight into its molecular mechanism. Currently there are several techniques available in attempting to find the optimal alignment of shared structural motifs between two proteins. Algorithms for the alignment of protein structures have grown increasingly important with the recent and rapid growth of the protein structure database. This paper analyzes protein structure alignment using dynamic programming and iterative improvement with algorithms. Protein structure alignment is, given two three-dimensional protein structures, to find spatially equivalent residue pairs. Research towards analysis of sequence–structure correspondences is critical for better understanding of a protein's structure, function and its interaction with other molecules. The proposed algorithms are shown to be useful through an experimental comparison with a previous alignment algorithm.

Keywords: Dynamic Programming (DP), Basic Alignment Algorithm (BASICALIGN), Dynamic Programming Alignment Algorithm (DPALIGN), Random Alignment Algorithm (RANDALIGN), Fragment-based Alignment Algorithm (FRAGALIGN), Root mean square distances (RMSD).

INTRODUCTION

Protein structure comparison has become an important bioinformatics tool for studying protein function and its evaluation[1,2]. Three-dimensional (3D) structures of proteins have been deposited in the Protein Data Bank[3] (http:/www.rcsb.org/pdb/) for easy reference for biological scientists. The goal of structure comparison is to relate proteins based on their structural similarity. 3D-structure of proteins is highly conserved than that of sequence based structure and structure alignment algorithms are frequently used to compare 3D-structure of proteins[4,5].

A variety of methods have been proposed for protein structure alignment[6-11]. Some scientists proposed iterative improvement methods [6-8] while others[9] developed a greedy method in which small fragments were assembled into larger structures. Taylor and Orengo developed the double dynamic programming technique.[10] Nussinov and Wolfson applied geometric hashing to protein structure alignment.[11] Moreover Sali and Overington developed a stochastic method using probability density functions.[12]. However, Holm et al. pointed out that each of these methods had one or more of limitations.[13] Moreover, most of these methods are not systematic but heuristic, and none of these methods have a theoretical guarantee for the quality of the obtained alignments. In computational geometry, a lot of studies have been done for geometric pattern matching problems.

However, most of them do not seem to be practical since they are too complicated and the time complexities are too high. Recently, Goodrich, Mitchell and Orletsky developed a practical algorithm for point set matching with a guaranteed approximation ratio.[14]

Although we use their technique in this paper, the protein structure alignment problem is more complex than their problem, and additional techniques are introduced in this paper.

METHODS

Problem Analysis

Here, we define the protein structure alignment problem in a formal way. First we consider representation of 3D protein structures. As we are only interested in representing an outline of 3D structure, we follow the common procedure of ignoring side chains and consider only C atoms (or the carbon and nitrogen atoms in the main chain), which are treated as points in 3D Euclidean space. Only the geometry of protein structures is considered and details such as the identity of specific atoms are ignored. Thus, each protein structure is treated as a sequence of points in 3D.

Next we define a distance d (P,Q) between point sequences where P = (p_1……p_n) and Q = (q_1……q_n)

by d(P,Q) = max||$\mathbf{p_i}$ $\mathbf{q_i}$ || where ||**xy**|| denotes the length of a line segment xy, Moreover, we define a distance D(P,Q) by D(P,Q) = mind(T(P),Q) where T takes any isometric transformation (rotation + translation) not including mirror image. Note that we can ignore mirror image without loss of generality because transformation including mirror image can be treated with increasing the computation time by a constant factor.[15]

In addition to d(P,Q) and D(P,Q), we use the root mean square distance (rms-distance, in short), which is widely used in molecular biology. Rms-distance d_{rms} (P, Q) between P and Q is defined by

$$d_{rms}(P,Q) = \min_{T} \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\overline{T(\boldsymbol{p}_i)\boldsymbol{q}_i}||^2}$$

Where T takes any isometric transformation, d_{rms} (P, Q) along with T can be computed in O (n) time using a simple method (a kind of least squares fitting method) .

For two point sequences P = (p_1....p_m) and Q = (q_1.....q_n), we call a partial correspondence

M= {(p_{i1}, q_{j1}),…,(p_{ik}, q_{jk})} between P and Q an alignment if i_1<i_2<....<i_k and j_1<j_2<.....<j_k hold.

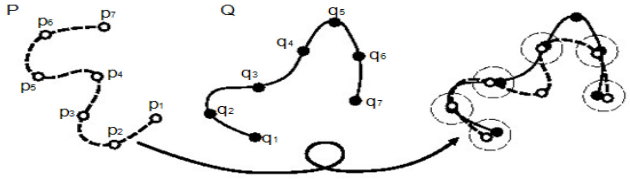

Fig. 1: Example of structural alignment

For an alignment of M, M(P) denotes a sequence(p_{i1},….,p_{ik}) of P and M(Q) denotes a subsequence (q_{j1},….,q_{j2}) of Q. Then we define the protein structure alignment problem in the following way:

Input

Point sequences $P = (p_1,....,p_m)$ and $Q = (q_1,....,q_n)$ and real number $\partial>0$.

Output

Alignment M along with transformation T which maximizes |M| under the condition that d(T(M(P)),M(Q))≸∂ if such M and T exist. Otherwise 'NO' is output.

Therefore, in this paper, we consider algorithms which approximately satisfy the condition d(T(M(P)),M(Q))<=∂.

Alignment of Protein Structure using FRAGALIGN Algorithm

Protein Structure alignment algorithm BASICALIGN, RANDALIGN and FRAGALIGN are implemented before. On these three, FRAGALIGN outputs best among them effectively and quickly. In this paper, we focused on FRAGALIGN algorithm. Although FRAGALIGN outputs effectively, here some simple modification are applied on FRAGALIGN based on transformation and ∂ to make the output more effective. This section describes total picture of FRAGALIGN Algorithm.

FRAGALIGN uses a very simple method to obtain initial super positions. Let Pi denotes a fragment $(p_i, p_{i+1},......, p_{i+L-1})$ of P, where L is a constant (L = 15 is used in current version). Qj is defined in the same way. Note that, for each pair of fragments Pi and Qj, we can obtain a superposition T (P)UQ using a transformation T which gives rmsd between Pi and Qj . FRAG tests initial super positions obtained from all pairs Pi and Qj in this way. Since there are O(mn) pairs and L can be considered as a constant, FRAG works in $O(m^2n^2)$ time. Although $O(m^2n^2)$ time is not efficient, the average case computation time can be reduced if we only test the cases where $d_{rms}(Pi,Qj)$ is small (for example, $d_{rms}(Pi,Qj) \leq 1.0A^o$).This implementation is heuristic, so FRAGALIGN (P,Q, ∂) does not miss good matching.[15]

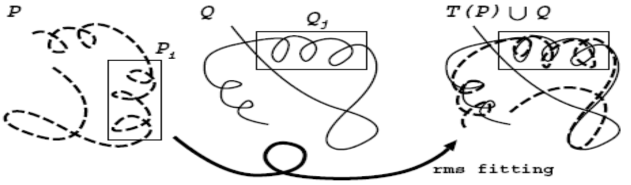

Fig. 2: Finding an initial superposition in FRAG

Existing FRAGALIGN Algorithm

```
Procedure FRAGALIGN (P, Q, $\partial$)
Begin
$M_1$:= {};
For I: = 1 to m-L+1 do
   For j: = 1 to n-L+1 do
            If d ($P_{i,I+L}$, $Q_{j,j+L}$)<=$\partial$ then
            Compute $T_{PP,QQ}$;
                Compute matching M = $M_{TPP,QQ}$(P,Q);
             If |M|> | $M_1$| then
Begin
$M_1$:= M; $T_1$:= $T_{PP,QQ}$
End
 End
End;
If $M_1$≠ {}; then Output $M_1$ and $T_1$
Else Output 'NO'
End.
```

Modified FRAGALIGN Algorithm

```
Procedure FRAGALIGN (P, Q, $\partial$)
Begin
$M_1$:= {};
Compute $T_{PP,QQ}$;
For i: = 1 to m-L+1 do
   For j: = 1 to n-L+1 do
If d($P_{i,i+L}$,$Q_{j,j+L}$)<=$\partial$ then
     Compute matching M = $M_{TPP,QQ}$(P,Q);
   If |M|> | $M_1$| then
  Begin
$M_1$:= M; $T_1$:= $T_{PP,QQ}$
End
 End
End;
If $M_1$≠ {}; then Output $M_1$ and $T_1$
Else Output 'NO'
End.
```

Difference between Two

Modifications are done to reduce the time complexity and rms distance between each pair of protein. Here at first protein transformations are performed before calculating the distance, two structure of protein are placed with the same origin (0,0,0),then length is selected. But in existing system distance is calculated before transformation. Next rms distance is calculated under the condition of d $(P_{i,i+L},Q_{j,j+L})<=\partial$,and paired each protein atom using cardinality matching .The time complexity is depends on constant factor ∂ and the computation steps. So the selection of constant factor ∂ is an important factor. To reduce the time complexity ∂ is selected here 2. Iterative improvement is avoided to reduce the implementation complexity.

RESULTS

FRAGALIGN were compared with a dynamic programming based algorithm (denoted by DP), in which input sequences are divided into small fragments and then a dynamic programming technique is applied.

Comparison has been done using PDB (Protein Data Bank) data and algorithms are implemented in C++ language. The experimental results are summarized in Table 1. Each item in DATA denotes a PDB code. The length (the number of points) is also described along with each structure. It is known that protein structures in the same row have similar structures. For each algorithm and each pair of structures, rmsd (d_{rms} (M (P), M (Q)) (A^o)) and the length (|M|) of the obtained alignment and CPU time (sec) are described. First observe that, in most cases, the rms distances obtained by RANDALIGN and FRAGALIGN (modified) are smaller than those by DPALIGN and the lengths of the alignments obtained by RANDALIGN and FRAGALIGN (modified) are longer than those by DPALIGN. Thus we can conclude that the proposed algorithms compute better alignments than DPALIGN. Next observe that the qualities of the alignments obtained by FRAGALIGN (modified) are as good as those by RANDALIGN, while the CPU times of FRAGALIGN (modified) are much shorter than those of RANDALIGN. Thus we can conclude that FRAGALIGN (modified) is more practical than RANDALIGN.

Table 1: Comparison of Structured Alignment Algorithms

Data		Dpalign			Randalign			Fragalign (modified)		
Data1 (P)	Data2 (Q)	Rmsd (A^o)	Len	Time (Sec)	Rmsd (A^o)	Len	Time (Sec)	Rmsd (A^o)	Len	Time (Sec)
1ubq	4fxc	2.54	40	1.03	2.22	58.9	4.94	1.81	50	0.3
3icb	5cpv	1.98	40	0.87	1.82	57.6	5.88	1.91	50	0.4
7cpy	1azu	2.89	50	0.33	2.34	71.9	30.52	1.94	74	0.7
4hhb	5mbn	2.18	300	28.55	1.13	324.9	70.66	1.13	91	2.0

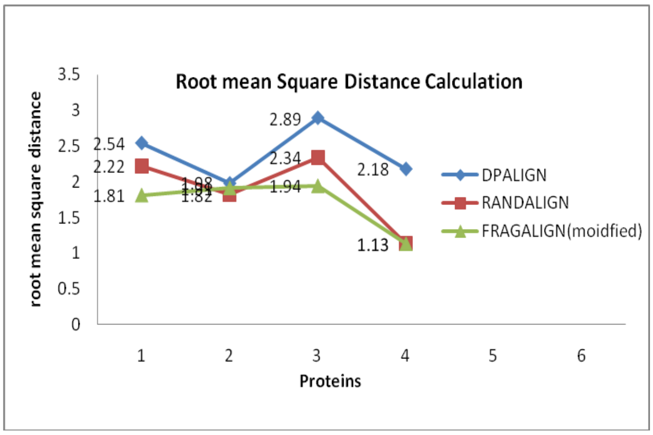

Fig. 3: Comparison graphs of Structured Alignment algorithms

In Table 2, we have compared modified FRAGALIGN results with existing FRAGALIGN results. These results show that the rmsd between points and the time to calculate rmsd and length are more less than existing system.

Finally in the table 3, we have compared FRAGALIGN (modified) output with TM-Align and FATCAT(Developed for Structure Alignment) output where rmsd of two proteins in FRAGALIGN (modified) are much lower than TM-ALIGN and FATCAT.

Table 2: Comparison of existing FRAGALIGN with modified FRAGALIGN

Data		Fragalign (existing)			Fragalign (modified)		
Data 1	Data 2	RMSD	Length	Time	RMSD	Length	Time
1UBQ	4FXC	2.35	57	0.32	1.81	50	0.3
3ICB	5CPV	1.78	58	0.55	1.91	50	0.4
7CPY	1AZU	2.30	71	0.82	1.94	74	0.7
4HHB	5MBN	1.50	114	2.40	1.13	91	1.2

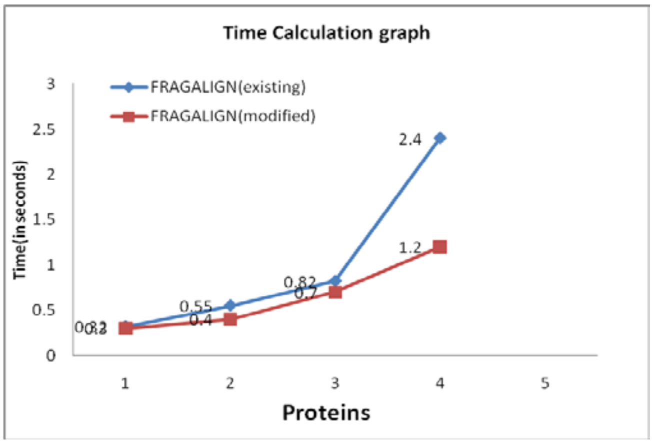

Fig 4. Time Calculation Graph

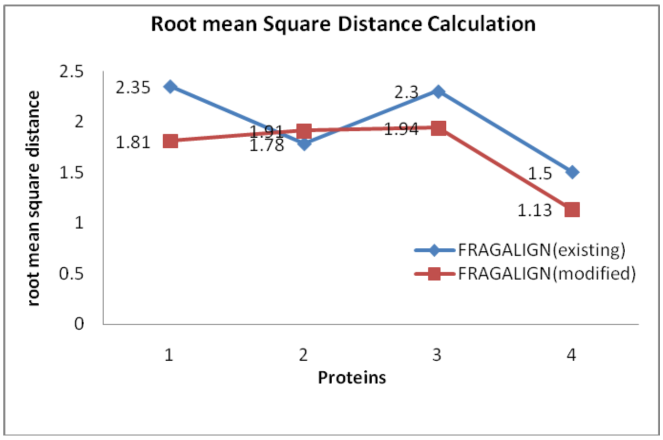

Fig 5. Root Mean Square Distance Calculation

Table 3: Comparison of TM-ALIGN, FATCAT and FRAGALIGN

Data		TM-Align		Fatcat		Fragalign (modified)	
Data1 (P)	Data2 (Q)	RMSD (A^0)	Len	RMSD (A^0)	Len	RMSD (A^0)	Len
1UBQ	4FXC	2.84	64	3.02	66	1.81	50
3ICB	5CPV	2.79	64	3.16	66	1.91	50
5CPV	5MBN	4.60	72	3.65	85	1.94	74
4HHB	5MBN	5.62	106	2.47	109	1.13	91

DISCUSSION AND CONCLUSION

Protein structure alignment is an important aspect in bioinformatics. We attempted here to design protein structure alignment algorithm. Although, very recently, we know that Holm and Sander have already proposed a structure alignment algorithm similar to ours in 1995.[16] They use an iterative improvement procedure almost same as ours. The difference between their algorithm and ours lies only in part of finding initial super positions. However, we have developed the proposed algorithms independently. Moreover, they do not give theoretical analysis and their algorithm does not have a guaranteed approximation ratio.

In this paper, we consider the protein structure alignment problem, which is a very important problem in molecular biology. Since an outline of protein structure is represented by a sequence of points in three dimensional space, this problem is defined as the following geometric pattern matching problem: given two point sequences P and Q in three dimensions and a real number $\delta>0$, find a maximum cardinality set of point pairs such that the distance between each pair is at most δ under the condition that any translation and rotation can be applied to P. Since it is very difficult to solve this problem exactly, we consider algorithms that solve it approximately. There are already developed algorithms: BASICALIGN, RANDALIGN and FRAGALIGN, whose worst case time complexities are $O(n^8)$, $O(n^5)$ and $O(n^4)$ respectively, where n denotes the size of larger input structure. All of these have the following common framework: a series of initial super positions are computed; for each of such super positions, a rough alignment is first computed using a dynamic programming technique, and then it is refined through an iterative improvement procedure which also uses dynamic programming; the best alignment among them is selected as an output. The difference among three algorithms lies in the methods of finding initial super positions. BASICALIGN, RANDALIGN and FRAGALIGN use exhaustive search, random sampling technique and fragment-based search, respectively.

There is a guaranteed approximation ratio (in the sense of distances between point pairs) for theoretical versions of BASICALIGN and RANDALIGN. In this paper, there is simple modification in FRAGALIGN algorithm which outputs improve previous FRAGALIGN outputs. Practical versions of RANDALIGN and modified FRAGALIGN are implemented and compared with a previous algorithm using real protein structure data. Modified FRAGALIGN is also compared with TM-ALIGN and FATCAT using real protein structure data. The experimental result of improved FRAGALIGN shows best among them and it outputs compute good alignments effectively and quickly. In Table 1, we have shown the comparison of structured alignment algorithms. In Table 2, we have compared modified FRAGALIGN results with existing FRAGALIGN results. These results show that the rmsd between points and the time to calculate rmsd and length are more less than existing system. In Table 3, we have compared FRAGALIGN (modified) output with TM-Align and FATCAT(Developed for Structure Alignment) output where rmsd of two proteins in FRAGALIGN (modified) are much lower than TM-ALIGN and FATCAT.

In this paper, we have proposed algorithms for protein structure alignment. Among them, theoretical versions of BASICALIGN and RANDALIGN have guaranteed approximation ratios. These are the first algorithms for protein structure alignment with guaranteed approximation ratios. Experimental results show that FRAGALIGN computes good alignments efficiently. Moreover, FRAGALIGN is simple and easy to implement. And with this simple modification FRAGALIGN becomes more practical. But there is a problem with these modifications which is length. Thus, to improve the algorithm for improving length calculation will be future work.

In this paper, each protein structure is treated as a rigid body. That is, alignments are computed considering global positions only. Although such a treatment is adequate for comparing structures with strong similarities, it is not adequate for comparing structures with weak similarities. Especially, in the case of classification of protein structures into the small number of families, finding weak similarities are important. Thus, to develop algorithms for finding weak similarities is important future work.

REFERENCES

1. Gibrat JF, Madej T, Bryant SH. Surprising similarities in structure comparison. Curr Opin Struct Biol 1996;6:377–385.
2. Holm L, Sander C. Searching protein structure databases has come of age. Proteins 1994;19:165–173.
3. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Res 2000;28:235–242.
4. Chothia C, Lesk AM. The relation between the divergence of sequence and structure in proteins. EMBO J 1986;5:823– 826.
5. Murzin AG. How far divergent evolution goes in proteins. Curr Opin Struct Biol 1998;8:380 –387.
6. Lesk AM. Protein Architecture: A Practical Approach, IRL Press, New York, 1991.
7. Pascarella S and Argos P. A data bank merging related protein structures and sequences. Protein Engineering. 1992; 5:121-137.
8. 8.Rao ST and M. G. Rossmann MG. Comparison of super-secondary structures in proteins. J. Mol. Biol. 1973;76:241-256.
9. Vriend G and Sander C. Detection of common three-dimensional substructures in proteins. PROTEINS: Structure, Function, and Genetics. 1991; 11:52-58.
10. Taylor WR and C. A. Orengo CA. Protein structure alignment. J. Mol Biol.1989;208:1-22.
11. Nussinov R and Wolfson HJ. Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. Proc. Natl. Acad. Sci. (USA). 1991;88 1049-10499.
12. Sali A and J. P. Overington JP. Derivation of rules for comparative protein modeling from a database of protein structure alignments. Protein Science. 1994;3:1582-1596.
13. Holm L, Onzounis C, Sander C, Tuparev G, and Vriend G. A database of protein structure families with common folding motif. Protein Science. 1992;1:1691-1698.
14. Goodrich MT, Mitchell JSB, and Orlet-sky MW. Practical methods for approximate geometric pattern matching under rigid motions. Proc. 10th ACM Symp. Computational Geometry.1994: 103-112
15. Akutsu T. Substructure search and alignment algorithms for three-dimensional protein structures. Research Report 94-AL-41-1, Information Processing Society of Japan, 1994:1-8
16. Holm L. and C. Sander C. 3-D lookup: fast protein structure database searches at 90% reliability. Proc. 3rd International Conference on Intelligent Systems for Molecular Biology (ISMB'95). 1995:179-187.