A Tool for Structure Alignment of Molecules

Pei-Ken Chang, Chien-Cheng Chen and Ming Ouhyoung
Department of Computer Science and Information Engineering, National Taiwan University
{zick, ccchen}@cmlab.csie.ntu.edu.tw, ming@csie.ntu.edu.tw

Abstract

In this paper, a novel tool is proposed to align two molecules (not just proteins) based on their 3D structural data, and the user can observe the result of alignment visually via the tool. Most existing tools are designed only for alignment of proteins. Here, a new tool is developed to address shared structural features between protein structures and tRNA structures, that is, molecular mimicry, although they are two very different types of molecules.

In order to align two molecules A and B, Geometric Hashing is applied to globally find initial matching of approximately overlapped atoms, thus parts of molecule A can be matched to parts of molecule B. Next, a fine tuning process is introduced, which is based on local optimization of overlapped parts, and the Iterative Closest Point (ICP) is used until the number of overlapped atoms within a given distance threshold can not be increased any more. The results show that our method is useful to structurally align two molecules, not restricted to align two proteins only. Besides, our tool outperforms in terms of RMSD and number of matched atom pairs in comparison to other tools.

1. Introduction

Search engines for 3D models have been developed in recent years [1] [2], however, can similar techniques be used in molecules? If so, the benefit can be great. The reason is that large number of protein structures can be determined by high throughput machines, classifying proteins into families and assigning functions to those novel proteins become major tasks in recent years. The Protein Data Bank (PDB) currently contains more than 25,000 structures and it is estimated that the number of structures in the PDB may exceed 35,000 by 2005. Though proteins have been grouped together on the basis of structural similarities in the FSSP [3], CATH [4], and SCOP databases [5], much effort still has been put into finding the similarities among proteins. Moreover, the rapid growth in the amount of structural data of proteins far exceeds the ability of experimental techniques to identify the locations and key amino acids of active sites. Although the structural genomics initiative (SGI) proposes to solve 10,000 protein 3D structures in this decade, however, many biological functions still remain unknown.

With the help of alignment tools, the structural similarity between proteins is revealed, as well as the functional and evolutional relationships. Holm and Sander [6] mentioned that structural similarities among distantly related proteins are often preserved in the process of evolution, but very little similarity at the sequence level.

There is an interesting problem studied, that is molecular mimicry. The molecular mimicry problem [7] is that a protein and a nucleic acid share a similar substructure, and sometimes it will even extend to similarity in interaction. Nissen et. al [8] indicated that the structure of Elongation Factor-G is similar to that of the complex of Elongation Factor-Tu and tRNA. Selmer et. al [9] mentioned that Ribosomal Recycling Factor looks like tRNA. In addition, exploitation of 3D structural data is a key factor to enhance structure-based drug design (SBDD), and the prediction of protein functions and possible active sites in proteins have become quite popular in SBDD, especially at front-ends to molecular docking [10] [11] or alternative active sites are sought otherwise.

This paper is organized as follows. Some related works are discussed in section 2. The geometric hashing algorithm and ICP algorithm we use are detailed in section 3. The experimental results are provided in section 4 while conclusion is given in section 5.

2. Previous Work

In general, structure alignment based on 3D structure has been shown to be NP complete by

Lathrop [12] and so heuristics are used to simplify the problem. Therefore, better methods for structure alignment are needed. Fisher et al. [13] used geometric hashing for a C_α-only representation of protein structure, and a follow-up is described in Tsai et al. [14]. Their method is based on preprocessing and recognition algorithms of complexity $O(n^3)$, where n is the number of residues of interest. Later, Pennec and Ayache [15] [16] introduced a 3D reference frame attached to each residue, which reduces the complexity of recognition to $O(n^2)$. Shindyalov and Bourne [17] proposed a method that involves a combinatorial extension (CE) of an alignment path defined by aligned fragment pairs (AFPs) rather than the more conventional techniques which use dynamic programming and Monte Carlo optimization. Combinations of AFPs that represent possible continuous alignment paths are selectively extended or discarded thereby leading to a single optimal alignment.

Zemla [18] proposed LGA (local-global alignment) algorithm, where longest continuous sequence is first found, and then a second step called GDT (global distance test) is applied. Both longest segment of residues under selected RMSD (root mean square distance) and largest set of equivalent residues that deviate less than a given distance threshold are obtained. Blankenbecler et al. [19] proposed to use fuzzy alignment variables and iterative minimization of a cost function. Milik et al. [20] used graph matching and represented atoms as nodes and bond distance as edge labels. The search method is based on comparison of local structure features of proteins that share a common biochemical function, and so does not depend on overall similarity of structures and sequences of compared proteins.

From the above survey, it is clear that all the above papers are concerned with proteins, and complexity reduction in alignment according to features of proteins or segments of aligned one dimensional sequence. Therefore, they can not solve the general molecule alignment problem unless the tools are modified.

3. Algorithms

In this paper, we propose a tool to align two molecules based on their 3D structural data. The alignment problem between two molecules A and B is solved in two steps: Geometric Hashing and a fine tuning process. Geometric Hashing globally finds initial matching of approximately overlapped atoms. Thus, parts of molecule A can be matched to parts of molecule B. Secondly, the fine tuning process is based on local optimization of overlapped parts, and the Iterative Closest Point (ICP) algorithm is used until the number of overlapped atoms within a given distance threshold can not be increased any more.

3.1. Geometric Hashing: Step One

Geometric hashing algorithm is introduced to structurally align two molecules. Geometric hashing algorithm is a technique originally developed in computer vision for object recognition and can easily be made parallel [21] [22]. In short, the geometric hashing algorithm is composed of two stages: *preprocessing* and *recognition*. The basic idea is to store in a database at preprocessing time a redundant representation of the models by rigid transformation. By doing so, the representation of the query object processed at recognition time will present some similarities with that of some database models. Matching is possible even when the recognizable database objects have undergone transformations or when only partial information is present.

Often the two interesting molecules are both proteins, so we will illustrate the solution in such a situation first. For some cases, e.g. molecular mimicry, two molecules belong to different type, there would be some variance while calculating, and we will describe later.

The three atoms N, C_α and C in each amino acid form a triangle which uniquely defines the position and orientation of the amino acid in the three-dimensional structure of a protein. Since the length of $N-C_\alpha$ and $C_\alpha-C$ are fixed, and $N-C_\alpha-C$ bond angle is also changeless. As alignment considered, the correspondence between two triplets of points in three-dimensional space is sufficient to uniquely determine a rigid transformation. With this mechanism, we can choose a single residue as a basis. A basis is calculated by the following steps and illustrated in Figure 1(a).

1. Normalize $\overline{NC_\alpha}$ to $\overline{e_1}$
2. $\overline{e_2} = \dfrac{\overline{e_1} \times \overline{C_\alpha C}}{\left|\overline{e_1} \times \overline{C_\alpha C}\right|}$
3. $\overline{e_3} = \overline{e_2} \times \overline{e_1}$

There are two phases, preprocessing and recognition, in the geometric hashing algorithm. To solve the problem of representation by different reference coordinates, coordinate information based on different reference frame of a model is encoded in the preprocessing phase and stored in a large memory, in this case, a hash table. The contents of the hash table are independent of the scene and thus can be computed offline to reduce the time needed for recognition. Accessing to the memory is based on geometric information that is invariant of the object's pose and computed directly from the scene. During the recognition phase, the method accesses the previously constructed hash table using the indices of the encoded coordinate information of the input object and finds their common spatial features.

In the phase of preprocessing, we calculate one basis for each residue to generate coordinates for each atom in a protein. In the phase of recognition, we choose a reference frame of the protein B. For each different reference frame of protein A in the hash table, we accumulate the number of matched atoms by checking whether there are two atoms close enough. We set a threshold distance *MatchThres* (*MatchThres* = 1 to 2Å is proper), beyond which atoms will not be considered as a match. If no atoms can be matched within *MatchThres*, we assign the score to 0. If there is an atom within *MatchThres*, we assign the score to 1. The process is repeated with each reference frame of the protein B until all the reference frames of these two proteins have been tested.

In the case of aligning two different kinds of molecules, the algorithm is slightly modified while creating the bases. For each atom whose coordinate is *P*, select two atoms connected with the atom, assuming that the coordinates for these two atoms are Q_1 and Q_2 respectively. The rule for constructing basis is

1. Normalize $\overline{PQ_1}$ to $\vec{e_1}$

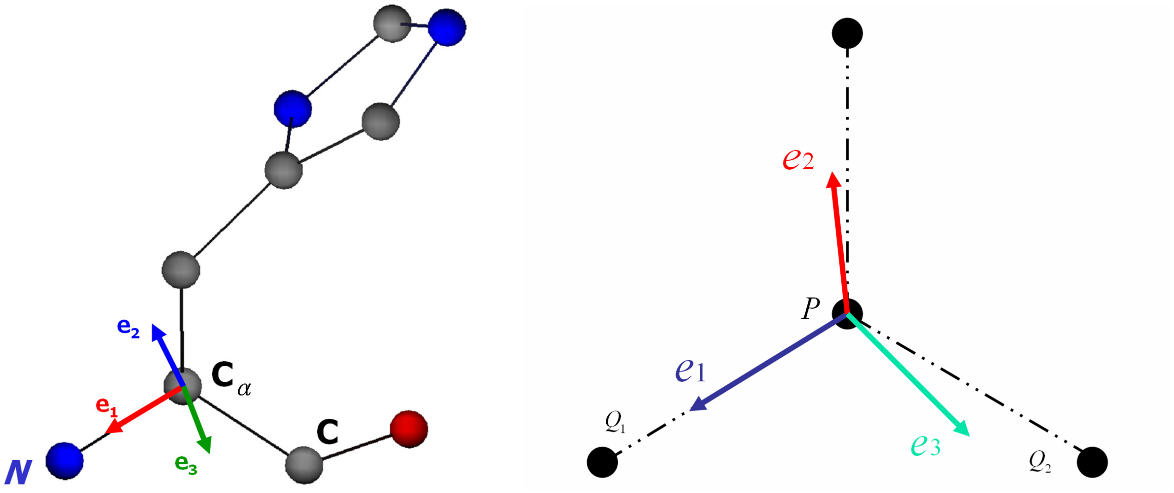
2. $\vec{e_2} = \dfrac{\vec{e_1} \times \overline{PQ_2}}{\left|\vec{e_1} \times \overline{PQ_2}\right|}$

3. $\vec{e_3} = \vec{e_2} \times \vec{e_1}$

and is illustrated in Figure 1(b). The origin of the new coordinate frame is P. If an atom is connected with n atoms, there would be $n \times (n-1)$ coordinate frames made for this atom. In this way, the number of constructed coordinate frames is too large so that the execution is not efficient. In order to decrease the execution time, the criteria for selecting atoms to create bases is listed in Table 1. Then we calculate two bases for each residue, while we calculate four bases for each nucleotide. In proteins, the " CA " atom is on the backbone and attached with a side-chain, and the " CB " atom is the attached atom. In nucleic acids, the " C4*" atom and the " C3*" atom are both on the similar position as the " CA " atom in proteins. And " O4*" atom and " C2*" are on the similar position as the " CB " atom in proteins. This is illustrated in Figure 2.

Table 1. The rule for selecting atoms to construct coordinate frames.

Type of the molecule	Name of the atom lie in *P*	Name of the atom lie in Q_1
Proteins	" CA "	" CB "
Nucleic Acids	" C4*"	" O4*"
	" C3*"	" C2*"



(a) (b)

Figure 1. Calculation of a basis. (a) The protein structure. (b) The general molecule structure.

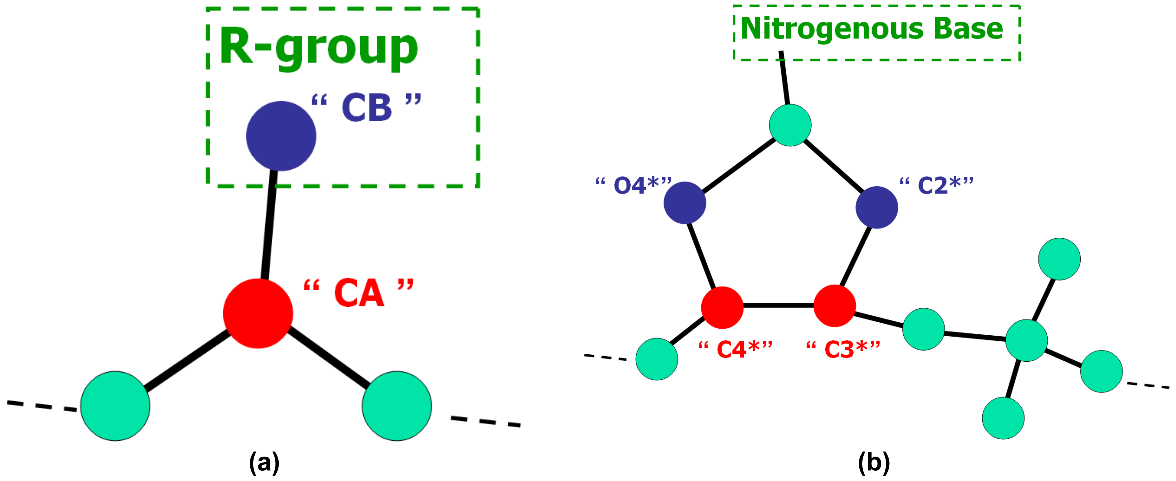


Figure 2. A sketch of molecules to explain the rule for coordinate frame construction. (a) Amino acid. (b)Nucleotide.

3.2. Fine Tuning Process: Step Two

Once the previous process is done by geometric hashing for global optimization with an output of approximate alignment, the following process is a fine tuning process based on local optimization of overlapped parts. This step is necessary, since the 3D structural data in PDB always involve sampling error in X-ray crystallography in determining atom positions. Furthermore, geometric hashing just provides initial alignment. Therefore the alignment needs fine tuning, and so Iterative Closest Point (ICP) algorithm [23] [24] is chosen. As illustrated in Figure 3, ICP algorithm is used in this process repeatedly, until the number of overlapped atoms within a given distance threshold can be increased no more.

The ICP algorithm proposes a solution to a key registration problem below: given two three-dimensional shapes, estimate the optimal translation and rotation that register the two shapes by minimizing the mean square distance between them. The algorithm guarantees that a local minimum of a mean square objective function is found [23]. In our implementation, we select 100 rigid transformations that lead to maximum numbers of overlapped pairs. The results show that ICP indeed increases the number of atoms matched.

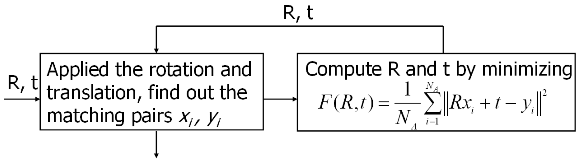

R, t
R, t
Applied the rotation and translation, find out the matching pairs x_i, y_i
Compute R and t by minimizing $F(R,t)=\frac{1}{N_A}\sum_{i=1}^{N_A}\|Rx_i+t-y_i\|^2$

Figure 3. The flow chart for fine tuning process.

4. Experimental Results

4.1. The Molecular Alignment Problem

Our tool can be used in solving the comparison of two molecules that belong to different types. The data and the problem of molecular mimicry (Figure 4, Figure 5) are provided by a graduate student Mr. Han Liang from Professor Laura Landweber's group in Dept. of Ecology and Evolutionary Biology, Princeton University [25].

One data set [8] consists of EFG (Elongation Factor-G) and EF-tu (the complex of Elongation Factor-Tu and tRNA), and the orientations of the original data are almost the same. The other data set [9] consists of RRF (Ribosomal Recycling Factor) and tRNA, but they are not in the same orientation originally. The aligning results of these two data sets are shown in Figure 6 and Figure 7.

After calculation by our tool, the rotation matrix between EFG and EF-tu/tRNA is

$$\begin{pmatrix} 0.993473 & -0.0945041 & -0.0638711 \\ 0.0832201 & 0.983483 & -0.160734 \\ 0.0780061 & 0.15437 & 0.984929 \end{pmatrix}$$

and the translation vector is $(-2.09762 \quad 0.935684 \quad -6.83966)$.

For the case of RRF and tRNA, the rotation matrix is

$$\begin{pmatrix} 0.42475 & 0.902835 & 0.0669089 \\ -0.545962 & 0.314407 & -0.776578 \\ -0.722159 & 0.293322 & 0.626458 \end{pmatrix}$$

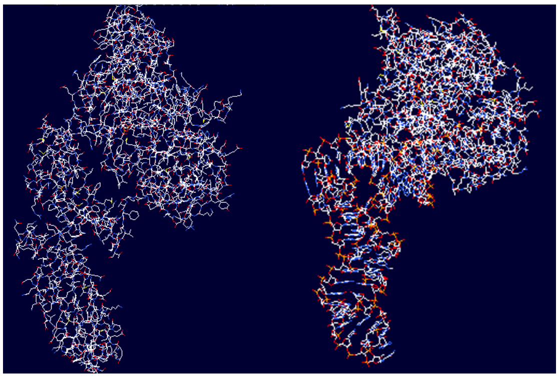


Figure 4. EFG vs. EF-tu/tRNA complex (Nissen et. al 1995 shows that the binding to ribosome is at the same place and orientation.) This picture is from Professor Laura Landweber's group of Ecology and Evolutionary Biology Dept. Princeton University, and the orientation is manually selected.

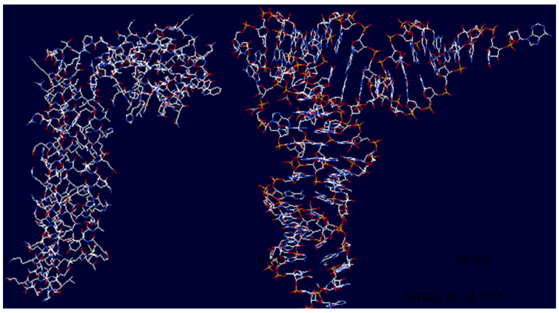


Figure 5. RRF vs tRNA (Selmer et. al 1999 shows that the binding to ribosome is at different place and orientation.), and again the orientation is manually selected.

and the translation vector is

$$\begin{pmatrix} -36.6722 & 65.1501 & 33.2718 \end{pmatrix}$$

4.2. Comparison with Other Alignment Tools

In order to compare with other tools, we will use the same set of proteins as in the paper of Blankenbecler et al. [19]. Note that other protein alignment methods usually use the knowledge of matched 1D sequence alignment for proteins, and they are optimized for proteins only focusing on backbone atoms C_α matching. Our tool does not have this assumption, and will work for arbitrary molecules, including tRNA. Still, for comparison purpose, we use the same set of six proteins. Figure 8 shows that our tool is better compared to other methods, where Figure 8(a) is reported from Blankenbecler's [19], in which Yale [26], Dali [27] [28], CE [17] and Lund [19] methods are compared, while Figure 8(b) is from our tool as compared to data in Figure 8(a).

The reasons why our method is better are

1. Given a fixed RMSD for pairs of matched atoms, our method has the most number of backbone C_α atoms;
2. Given fixed number of matched C_α, our method has the lowest RMSD.

In terms of computation cost, the major cost is in the first step, the geometric hashing. In the case of proteins, the coordinate frames are generated from the amino acid C_α atoms only, and thus the computation cost is low. For the six pairs of target proteins, all alignment calculation is done ranging from 6 seconds to 47 seconds. Table 2 shows the computation time on a Pentium-4 3GHz PC.

In the case of molecules such as RNA and DNA, the nucleic acid has a carbon ring in its base, and therefore the number of possible coordinate frames tends to be much more than that of proteins. Certainly, the computation time is longer. In the case of RRF vs. tRNA, where there are over 1000 atoms in tRNA, the computation time is around 24 minutes, while in the case of EFG vs. EF-tu/tRNA complex (over 4000 atoms), the computation time can be as long as 36 hours on the same 3 GHz PC. Even so, our tool can still solve this problem, which is a very important problem called "molecular mimicry". As far as we know, our method is the first one to solve this kind of problems, because our algorithm is sequence independent, and does not use the knowledge of 1D sequence similarity in molecule pairs.

5. Conclusion

A novel tool is developed to align two molecules based on 3D structural data. In contrast to other algorithms, it takes more computation time to align two molecules by our tool. However, other tools might be restricted to align two proteins. The experiments are conducted based on the data from the PDB and demonstrate that the proposed tool is useful and versatile.

The first experiment is the molecular alignment problem. Given two molecules, our tool will generate the rotation matrix and translation vector so that the above two molecules are optimally aligned. In our experiments, the results are the same, no matter where we randomly place the molecules in a different location with different orientation.

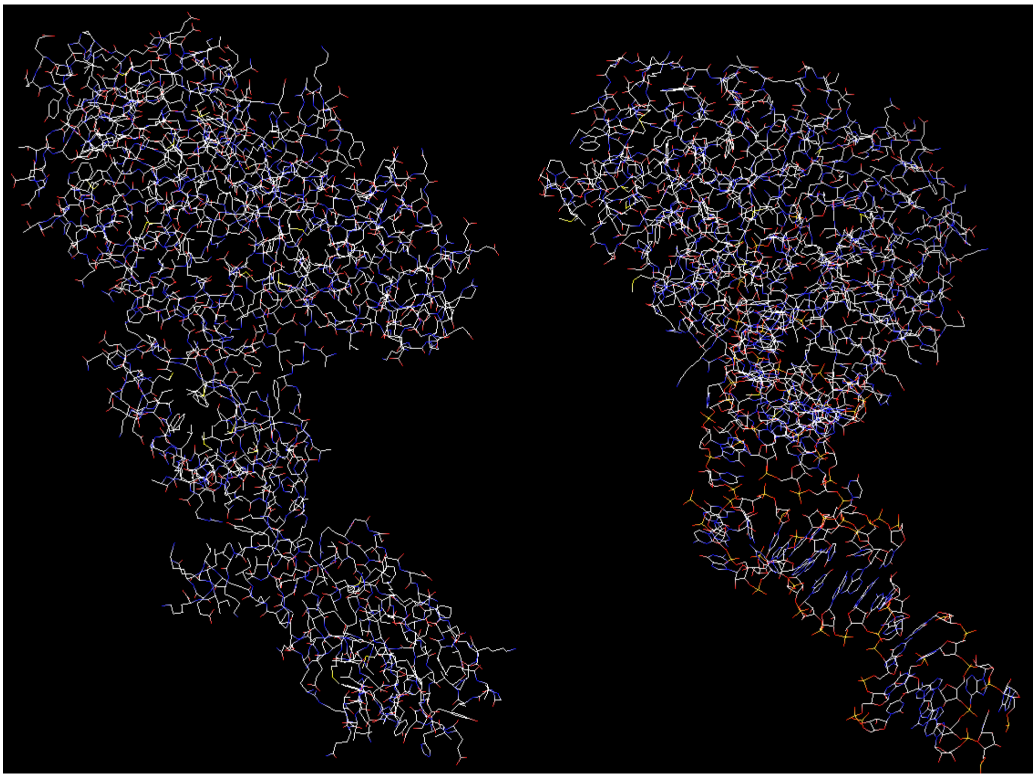


Figure 6: Alignment of two molecules using our tool for EFG vs. EF-tu/tRNA complex, where the atom number is over 4000 and the computation time is about 36 hours on a Pentium-4 3GHz PC.

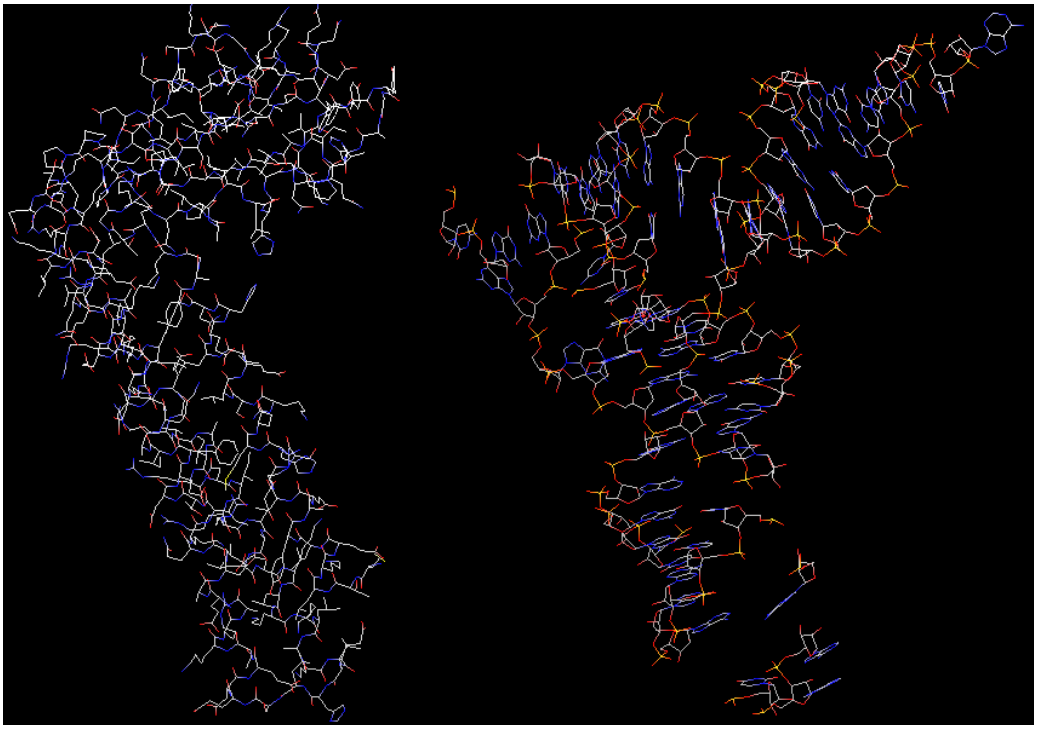
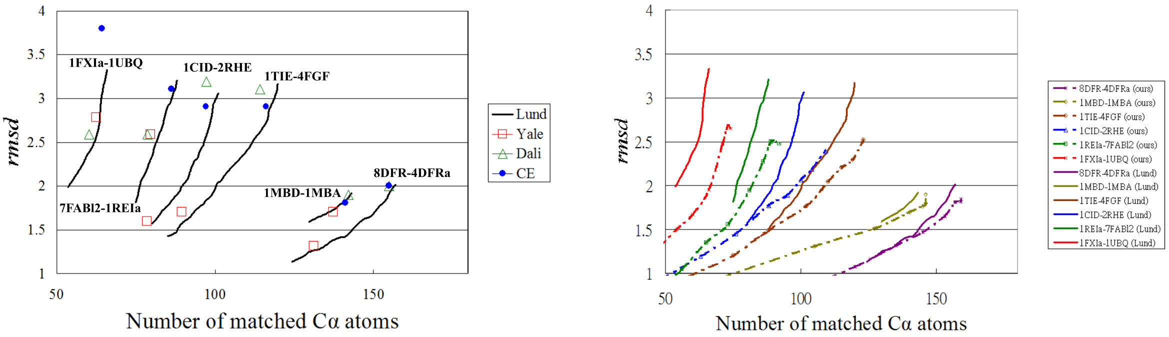


Figure 7: Alignment of two molecules using our tool for RRF vs. tRNA, where the atom number is over 1000 and the computation time is about 24 minutes on a Pentium-4 3GHz PC.



(a) **(b)**

Figure 8: Alignment results for a set of protein pairs in terms of RMSD of matched atom pairs and number of aligned atoms (N). In this figure, (a) is from Blankenbecler et al. fuzzy alignment method. The results from Yale (red squares), Dali (green triangles), CE (blue circles), and Lund method (solid lines) are also given in their paper. (b) is from our tool as a comparison. It shows that our results are better as compared with other methods.

Table 2: Computation time of alignment of six pairs of proteins, where *MatchThres* means the threshold used in initial geometric hashing, while the other columns are in seconds.

MatchThres (Å)	8DFR-4DFRa	1MBD-1MBA	1TIE-4FGF	1CID-2RHE	7FABl2-1REIa	1FXIa-1UBQ
1.0	7	5	4	3	2	1
1.5	9	6	5	5	2	1
2.0	11	7	6	5	3	1
2.5	13	10	8	7	3	2
3.0	18	12	9	9	4	3
3.5	22	16	13	11	5	3
4.0	30	20	15	13	7	3
4.5	37	26	20	18	8	6
5.0	47	34	24	22	10	6

In the second experiment, several protein pairs are used to compare the results with four popular alignment tools, namely Yale [26], Dali [27] [28], CE [17] and Lund [19] methods. Our tool performs the best in terms of RMSD and number of matched atom pairs.

6. References

[1] D.Y. Chen, X.P. Tian, Y.T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval" *Comput. Graph. Forum, 22(3)*, 2003, pp. 223-232.

[2] T. Funkhouser, P. Min, M. Kazhdan, J. Chen , A. Halderman, D. Dobkin, and D. Jacobs, "A search engine for 3d models" *ACM T. Graphics, 22(1)*, Jan. 2003, pp. 83-105.

[3] L. Holm and C. Sander, "Touring protein fold space with Dali/FSSP" *Nucl. Acids Res., 26*, 1998, pp. 316-319.

[4] C.A. Orengo, A.D. Michie, S. Jones, D.T. Jones, M.B. Swindells, and J.M. Thornton, "CATH - a hierarchic classification of protein domain structures", *Structure, 5(8)*, Aug. 1997, pp. 1093-1108.

[5] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins for the investigation of sequences and structures", *J. Mol. Biol., 247*, 1995, pp. 536-540.

[6] L. Holm and C. Sander, "Mapping the protein universe", *Science, 273*, Aug. 1996, pp. 595-602.

[7] P. Nissen, M. Kjeldgaard, and J. Nyborg, "Macromolecular mimicry", *EMBO J., 19*, 2000, pp. 489-495.

[8] P. Nissen, M. Kjeldgaard, S. Thirup, G. Polekhina, L. Reshetnikova, B.F.C. Clark, and J. Nyborg, "Crystal structure of the ternary complex of Phe-tRNA^{Phe}, EF-Tu, and a GTP analog", *Science, 270*, Dec. 1995, pp. 1464-1472.

[9] M. Selmer, S. Al-Karadaghi, G. Hirokawa, A. Kaji, and A. Liljas, "Crystal structure of Thermotoga maritima ribosome recycling factor: a tRNA mimic", *Science, 286*, Dec. 1999, pp. 2349-2352.

[10] V. Cappello, A. Tramontano, and U. Koch, "Classification of proteins based on the properties of the

ligand-binding site: the case of adenine-binding proteins", *Proteins, 47(2)*, May 2002, pp. 106-115.

[11] G.R. Smith and M.J. Sternberg, "Prediction of protein-protein interactions by docking methods", *Curr. Opin. Struct. Biol., 12(1)*, Feb. 2002, pp. 28-35.

[12] R.H. Lathrop, "The protein threading problem with sequence amino acid interaction preferences is NP-complete", *Protein Eng., 7*, 1994, pp. 1059- 1068.

[13] D. Fischer, O. Bachar, R. Nussinov, and H. Wolfson, "An efficient automated computer vision based technique for detection of three dimensional structural motifs in proteins", *J. Biomol. Struct. Dyn., 9(4)*, Feb. 1992, pp. 769-789.

[14] C.J. Tsai, S.L. Lin, H. Wolfson, and R. Nussinov, "Techniques for searching for structural similarities between protein cores, protein surfaces and between protein-protein interfaces", *Techniques in Protein Chemistry, VII*, 1996, pp. 419-429.

[15] X. Pennec and N. Ayache, "An O(n^2) algorithm for 3D substructure matching of proteins", *Shape and Pattern Matching in Computational Biology - Proc. First Int. Workshop*, 1994, pp. 25-40.

[16] X. Pennec and N. Ayache, "A geometric algorithm to find small but highly similar 3D substructures in proteins", *Bioinformatics, 14(6)*, 1998, pp. 516-522.

[17] I.N. Shindyalov and P.E. Bourne, "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path", *Protein Eng., 11(9)*, Sep. 1998, pp. 739-747.

[18] A. Zemla, "LGA: A method for finding 3D similarities in protein structures", *Nucleic Acids Res., 31(13)*, Jul. 2003, pp. 3370-3374.

[19] R. Blankenbecler, M. Ohlsson, C. Peterson, and M. Ringner, "Matching protein structures with fuzzy alignments", *Proc. Natl. Acad. Sci. USA., 100(21)*, Oct. 2003, pp. 11936-11940.

[20] M. Milik, S. Szalma, and K.A. Olszewski1, "Common structural cliques: a tool for protein structure and function analysis", *Protein Eng., 16(8)*, Aug. 2003, pp. 543-552.

[21] Y. Lamdan and H.J. Wolfson, "Geometric hashing: a general and efficient model-based recognition scheme", *Proceedings of the Second ICCV*, 1988, pp. 238-249.

[22] H.J. Wolfson and I. Rigoutsos, "Geometric hashing: an overview", *IEEE comp. Science and Eng., 4*, 1997, pp. 10-21.

[23] P.J. Besl and N.D. McKay, "A method for registration of 3-D shapes", *IEEE T. Pattern ANAL., 14*, 1992, pp. 239-256.

[24] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces", *Int. J. Comput. Vision, 13(2)*, 1994, pp. 119-152.

[25] H. Liang and L.F. Landweber, "Computational tests of molecular mimicry between tRNA and protein translation factors", submitted, 2004.

[26] M. Gerstein and M. Levitt, "Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures", *Proc. Int. Conf. Intell. Syst. Mol. Biol., 4*, 1996, pp. 59-67.

[27] L. Holm and C. Sander, "Protein structure comparison by alignment of distance matrices", *J. Mol. Biol., 233*, 1993, pp. 123-138.

[28] L. Holm and J. Park, "DaliLite workbench for protein structure comparison", *Bioinformatics, 16*, 2000, pp. 566-567.