Tools and Algorithms for Refined Comparison of Protein Structures

YAW-LING LIN*
Providence University
Dept. Comput. Sci. & Info. Engineering
200 Chung Chi Road, Shalu, Taichung 433
TAIWAN

SHIH-PENG HUANG
Providence University
Dept. Comput. Sci. & Info. Management
200 Chung Chi Road, Shalu, Taichung 433
TAIWAN

Abstract: Protein structure provides the opportunity to recognize homology that is undetectable by sequence comparison, and it represents a powerful means of discovering functions, yielding direct insight into the molecular mechanisms. In this paper, we propose algorithms and develop tools for pairwise alignment of protein structures. Methods of locating suitable isometric transformations of one structure and aligning it to the other structure are addressed. Our methods allow sequence gaps of any length, reversal of chain direction, and free topological connectivity of atom sequences. We show the effectiveness of the proposed refinement methods by a set of experiments, which improve several previous results.

Key–Words: structural proteomics, algorithms, structure alignments and comparisons, rmsd.

1 Introduction

One of the primary goals of structural alignment programs is to quantitatively measure the level of structural similarity between all pairs of known protein structures. This data can provide several meaningful insights into the nature of protein structures and their functional mechanisms.

The three dimensional structure of proteins is highly conserved during evolution [3]. Protein are constructed by one or more polypeptide chains that fold into complicated 3D structures. Detection of proteins with a similar fold can suggest a common ancestor, and often a similar function [5]. Comparison of 3D structures makes it possible to establish distant relationships, even between protein families distinct in terms of sequence comparison alone. This is why structural alignment of proteins increases our understanding of more distant evolutionary relationships [2, 9]. The link between structural classification and sequence families enables us to study functions of various folds, or whole proteins [10].

The smallest *root mean squared deviation (rmsd)* is a least-squares fitting method for two sequences of points [8]. The idea is to align atom vectors of the two given (molecular) structures, and use the common least averaged squared errors as a measurement of differences between these two (paired) sequences. Formally, let $P = \langle p_1, \ldots, p_n \rangle$ and $Q = \langle q_1, \ldots, q_n \rangle$ be two sequences of points. We assume that P is translated so that its centroid $(\frac{1}{n}\sum_{k=1}^{n} p_k)$ is at the origin. We also assume that Q is translated in the same way. For each point or vector x, let $(x)_i (i = 1, 2, 3)$ denote the i-th (X, Y, Z) coordinate value of x, and $\|x\|$ denote the length of x. Let

$$\text{RMSD}(P, Q, R, \mathbf{a}) = \sqrt{\frac{1}{n}\sum_{k=1}^{n} \|Rp_k + \mathbf{a} - q_k\|^2} \tag{1}$$

where R is a rotation matrix and $\mathbf{a}$ is a translation vector. Then, the *rmsd* value $d(P, Q)$ between P and Q is defined by $d(P, Q) = \min_{R,\mathbf{a}} d(P, Q, R, \mathbf{a})$. Although complicated as it might appear, the optimal rotation matrix and translation vector can be found simultaneously in $O(n)$ time. Schwartz [14] showed that $d(P, Q, R, \mathbf{a})$ is minimized when $\mathbf{a} = 0$ and

$$R = (A^t A)^{\frac{1}{2}} A^{-1} \tag{2}$$

where the matrix $A = (A_{ij})\ i, j = 1, 2, 3$ is given by

$$A_{ij} = \sum_{k=1}^{n} (p_k)_i (q_k)_j \tag{3}$$

, $A^{\frac{1}{2}} = B$ means $BB = A$, and $\mathbf{o}$ denotes the zero vector. Thus, $d(P, Q)$, R and $\mathbf{a}$ can be computed in $O(n)$ time [12].

*This work is supported in part by grants from the Taichung Veterans General Hospital and Providence University (TCVGH-PU- 958101) and in part by the National Science Council (NSC-95- 2221-E-126-007), Taiwan, R.O.C.

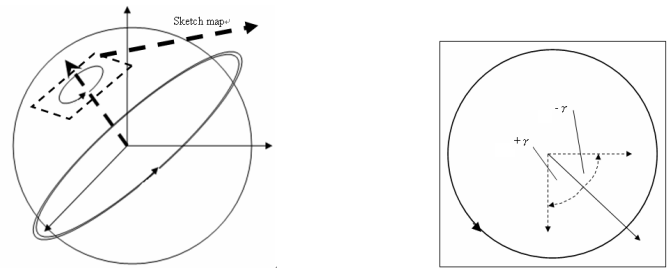

Figure 1: The movement of a (parametric) probe.

We adopt Martin's ProFit (for protein fitting system) [11] to calculate the RMSD between Cα atoms of paired protein backbones. ProFit has many features including flexible specification of fitting zones and atoms, calculation of RMS over different zones or atoms, RMS-by-residue calculation. Fitting was performed using the McLachlan algorithm [12].

There have been several methods proposed to compare protein structures and measure the degree of structural similarity based on alignment of secondary structure elements as well as alignment of intra and inter-molecular atomic distances. The basic ideas are rapid identification of pair alignments of secondary structure elements, clustering them into groups, and scoring the best substructure alignment. For examples, the VAST system is based on continuous distribution of domains in the fold space. The FSSP/DALI system provides two levels of description – a coarse-grained one and one with a fine-grained resolution. The method, CATH, provides the complete PDB fold classification by domains and links to other sources of information. The two methods, CE and LGscore2, are based on a different idea. They focus on the local geometry rather than global features such as orientation of secondary structures and overall topology (as in the case of VAST or DALI) [4, 8, 15]. VAST has been used to compare all known PDB domains to each other. The results of this computation are included in NCBI's Molecular Modelling Database at `http://www.ncbi.nlm.nih.gov/Structure/-VAST/vast.html`.

Note that there must be an atom-pairing scheme before one can do the *rmsd* computation. The first atom of the first selection is compared to the first atom of the second selection, fifth to fifth, and so on. Our objective in this paper is to calculate the significance of score (rmsd) between spatial arrangements of Cα atom of protein backbone that are not necessarily adjacent in sequence. By matching the backbone Cα atoms between two sets of atoms, the algorithm can obtain lower (rmsd) scores comparing to these existed protein structure alignment systems like VAST or DALI.

2 Method

Consider the point of north-pole $\mathbf{n} = (0, 0, 1)$ on the unit sphere. After the rotation, R, say $\mathbf{n}$ is rotated to another point $\mathbf{p} = (x, y, z)$; i.e., $\mathbf{p} = R\mathbf{n}$. Let α denote the angle $\angle \mathbf{n}O\mathbf{p}$. Note that α determines the z-coordinate of $\mathbf{p}$. To determine x-coordinate and y-coordinate of $\mathbf{p}$, the point is rotated around the z-axis for the angle β on the unit sphere. Note that there are infinitely numbers of rotation that transform $\mathbf{n}$ to $\mathbf{p}$. The particular rotation R can be decided by rotating all other points around the vector $\mathbf{p}$ by the angle γ. It is not hard to verified that, in such a way, *any* rigid rotation transformation can be parameterized by the three-tuple (α, β, γ). Thus, we call a vector $\mathbf{p} = (x, y, z)$ on the surface of the unit sphere a *probe*. Note that the *movement* of each probe is started from the north-pole $(0, 0, 1)$ to other points in the sphere. The position of $\mathbf{p}$ is decided by the parameters (α, β), and exact rotation is fixed by the self-rotation angle γ. The movement of the probe is illustrated by Figure 1.

The main idea of our algorithm for finding a suitable matching between two sets of points before utilizing the RMSD procedure to fine-tune the final result is by searching the suitable (parametric) probe. After that, we use the minimum bipartite matching algorithm to find the best matching between two sets to decide the best matching for the RMSD procedure. Let $P' = T \circ P$, and Q being translated to Q' such that the mass center of Q' is located at the origin. We construct a weighed graph $G = (V, E)$ with V being labelled with points of P' and Q', and each (p, q)

ROT-MAT$(\mathbf{r}, \theta)$
Input: A normal vector $\mathbf{r} = (x, y, z)$ and rotation angle θ
Output: A 3×3 rotation matrix M
1 Compute $c \leftarrow \cos\theta$; $s \leftarrow \sin\theta$
2 Return the rotation matrix $M =$

$$\begin{bmatrix} c + x^2 \cdot (1-c) & xy \cdot (1-c) - zs & xz \cdot (1-c) + ys \\ xy \cdot (1-c) + zs & c + y^2 \cdot (1-c) & yz \cdot (1-c) - xs \\ xz \cdot (1-c) - ys & yz \cdot (1-c) + xs & c - z^2 \cdot (1-c) \end{bmatrix}$$

Figure 2: The matrix ROT-MAT rotates 3D points around the vector $\mathbf{r}$ by angle θ.

in E being weighted with the squared Euclidean (3D) distance; i.e., $w(p, q) = \|p, q\|^2$. We then solve the weighted minimum bipartite matching problem [6] to obtain the best matching of P' and Q'. By the matched pairing, we perturb and refine the final alignment to obtain lower *rmsd*.

Since the best parametric probe will be very difficult to locate, the idea is to send out a team of several probes over the sphere and let each probe searching its own proximity in a randomized greedy manner. Note that the best rotation R consists of two part including the location of the probe and the self-rotating angle. In the following, Section 2.1 explains the procedure that a probe searching better proximity in reducing its *rmsd* value. Section 2.2 discusses the procedure to spread n probes uniformly on the sphere at random in details.

2.1 Probes searching on the sphere

For a probe to search better minimum matching and obtain better *rmsd* values in its proximity, it needs to try out another possible location on the sphere. The ROT-MAT procedure can be used to reach the goal. For a probe $\mathbf{p}$, we can locate in its neighborhood another possible location $\mathbf{p}'$ to be its new location on the sphere before performing another self-rotating by an angle γ. Note that the corresponding rotation matrix $M = R(\mathbf{p}, \mathbf{p}')$ can be obtained by the formula $M =$ ROT-MAT$(\mathbf{p} \times \mathbf{p}', \angle \mathbf{p}o\mathbf{p}')$, as shown in Figure 2.

It remains to show how to locate the neighboring point $\mathbf{p}'$ uniformly distributed over the open disc around $\mathbf{p}$ on the sphere. Let $f(x)$ be the probability density function of a random variable $X : \mathbb{R} \rightarrow \mathbb{R}$, and let F be the probability cumulative function $F(x) = \int_{-\infty}^{x} f(t)dt$. It can be verified that $x = F^{-1}(p)$, where p is taken uniformly from $(0, 1)$, is a reasonable way of generating a random sample x that matches the desired probability distribution defined by f.

It follows that the neighboring point $\mathbf{p}'$ can be obtained by first deciding the rotating angle around $\mathbf{p}$. Let RAND() denote a real-valued random function uniformly distributed over $(0, 1)$. Clearly we expect that $\theta = 2\pi \cdot$ RAND(). Further, we need to decide the distance d from $\mathbf{p}'$ to $\mathbf{p}$. Since we expect that $\mathbf{p}'$ is uniformly distributed over the open disc around $\mathbf{p}$ on the sphere, it follows that Prob$\{\|\mathbf{p}, \mathbf{p}'\| = d\} \propto d$ when $\mathbf{p}'$ lies inside the disc. Let r be the radius of the predefined open disc. Note that the cumulative function $F(d) = d^2/r^2$ in this case. Thus the distance d can be obtained by the formula $d = r \cdot \sqrt{\text{RAND()}}$.

2.2 Uniformly spreading n probes

To uniformly place n probes on surface of the sphere, we consider the distribution function of the location of a probe $\mathbf{p}$ on the sphere in terms of the two parameters α and β. Note that β is clearly uniformly distributed in the range $(0, 2\pi)$. On the other hand, α is not. It is not hard to verify that $f(\alpha) \propto \cos\alpha$; thus $F(\alpha) = \sin\alpha$. It follows that $\alpha = \sin^{-1}(\text{RAND()})$, is a way of generation a good random sample point. It is interesting to note that, since $z = \sin(\alpha)$, we can just set $z =$ RAND() to be the desired distribution. The detailed procedure is shown in Figure 3.

3 Experiments and Result

We have implemented these algorithms, by incorporating serval existing systems as well as writing thousands lines of C codes in the Linux environment. In particular, we improve our previously developed system [10], where the minimum weighted bipartite matching algorithm was adapted from the LEDA [13] package, where the matching algorithm is implemented by Dijkstra's algorithm as heuristics in adding

RAND-SPHERE-POINT()
Output: A vector $\vec{r}(x, y, z)$ uniformly placed on the sphere
1 $z \leftarrow$ RAND(); **if** RAND() < 0.5 **then** $z \leftarrow -z$
2 $\theta \leftarrow 2\pi \cdot$ RAND()
3 return $(\cos\theta, \sin\theta, z)$

Figure 3: Generating a random point that is uniformly distributed on the unit sphere.

Paired moleculars $(M_1 : M_2)$	VAST *rmsd* (A)	Improved *rmsd* (B)	Lin[10] *rmsd* (C)	Improved ratio (%) $(A-B)/A$	Improved ratio (%) $(C-B)/C$
101M:2DHB-B	1.66	1.61	1.62	2.78	0.62
101M:1CH4-A	1.49	1.45	1.44	2.82	-0.69
1MLL:1HLM	2.07	1.98	2.08	4.26	4.81
102M:1KFR-A	2.31	2.27	2.30	1.52	1.30
102M:1SPG-A	1.67	1.61	1.61	3.71	0
1SPG-A:1H1X-A	1.76	1.71	1.71	2.84	0
1SPG-A:1SCT-A	2.13	2.09	2.12	1.83	1.42
3HHB-A:1RSE	1.69	1.64	1.64	2.73	0
3HHB-A:1HRM	1.82	1.76	1.76	3.19	0
3HHB-A:1DM1-A	2.27	2.17	2.21	4.36	1.81
2DHB-A:2MGF	1.57	1.53	1.52	2.30	-0.66
2DHB-A:1RSE	1.69	1.64	1.64	2.73	0
1OUT-A:1MOC	1.76	1.69	1.73	3.76	2.31
1OUT-A:1CH2-A	1.74	1.69	1.71	2.99	1.17
1H1X-A:1CH4-A	1.63	1.6	1.61	1.72	0.62
1H1X-A:1FHJ-B	1.67	1.67	1.67	0	0
1H1X-A:1O1P-B	1.80	1.79	1.79	0.83	0

Figure 4: Improvement ratios of our algorithm.

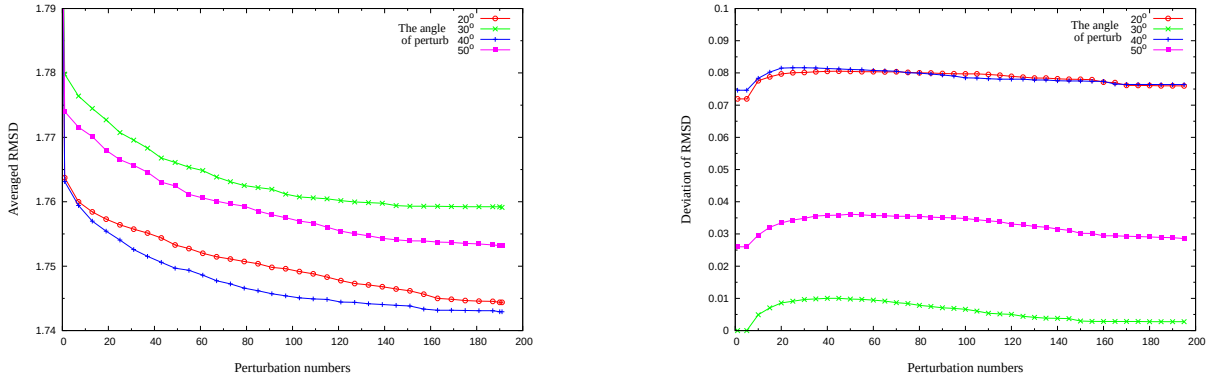

Figure 5: The progress of *rmsd* under different $\theta_{\max} = 20°, 30°, 40°, 50°$, fixing $\gamma_{\max} = 20°$.

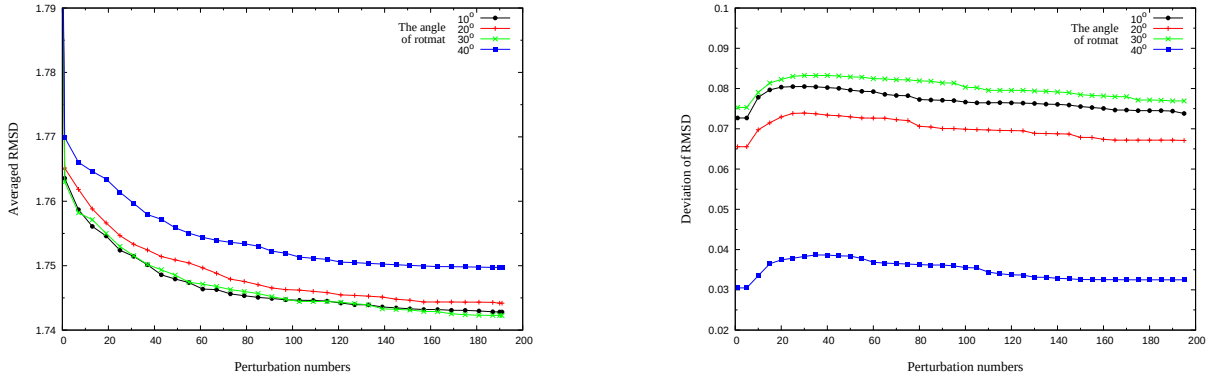

Figure 6: The progress of *rmsd* under different $\gamma_{\max} = 10^\circ, 20^\circ, 30^\circ, 40^\circ$, fixing $\theta_{\max} = 30^\circ$.

the augmenting path. In the worst case, the time complexity of this algorithm is $O(n(m+n\log n))$ [13]. In the current version of the system, we adapt the local-improvement methodology, and the improved version generally runs with better efficiency, since most of the time, the probe usually gradually improves its previous found *rmsd* by some local exchanges of some pairings, where the new method is taking advantage of.

The experiments demonstrate that our structure alignment algorithms find better results comparing to some available structure comparison methods; e.g., NCBI's Vector Alignment Search Tool (VAST) [7]. Several sets of real protein structures are randomly picked from the PDB [1] for comparing the effectiveness of our algorithms. We have selected the same groups of proteins with previous experiments [10] to compare these results. The result is shown in Figure 4.

In summary, for these (randomly selected) 17 pairs of PDB protein samples shown in Figure 4, our current structure comparison system improved the quality of VAST's *rmsd* by about 2.59% in average. Comparing to our previous results, the current system generally even further improves the results by about 0.75% in average.

Note that the methodology of the current system is generally a randomized algorithm whose performance is depended on various setting of the control parameters. For example, the number of initially probes on the sphere, the radius of the open disc surrounding the probe, the maximum angle of a self-rotation probe each time, and so on. In order to obtain suitable setting of parameters, we focus on the samples discussed above, and carry experiments to find out good setting parameters of the systems. Since the randomized features, for each data and each parameter settings, we perform the experiments over 20 times and analyze the averaged performance together with the standard deviation.

The first experiment tries to analyze how the movement (speed) of probes affects the final found *rmsd*. The decisive parameter is the radius of the open disc on the sphere. Note that the radius can be viewed from the center as the maximum rotating angle. Assuming that the maximum self-rotating angle $\gamma_{\max}$ is fixed to 20°, we test the parametric settings by analyzing the maximum angle of probe movement, $\theta_{\max}$ of $20^\circ, 30^\circ, 40^\circ$, and 50°. The relation between the perturbation numbers and *rmsd* is drawn in the diagram of Figure 5. The second experiment tries to analyze how the maximum self-rotating angle $\gamma_{\max}$ affects the final *rmsd*. Assuming that the maximum angle of probe movement $\theta_{\max}$ is fixed to 30°, we test the maximum rotating angle $\gamma_{\max}$ of $10^\circ, 20^\circ, 30^\circ$, and 40°. The relation between the perturbation numbers and *rmsd* is drawn in the diagram of Figure 6.

In summary, both experiment 1 and experiment 2 shows that the system gradually obtains better (smaller) *rmsd*'s when the number of perturbation process increase. Furthermore, it is observed that suitable setting for the maximum angle of probe movement, $\theta_{\max}$ would be about 20° or 40°, while a good setting for the maximum self-rotating angle $\gamma_{\max}$ would be about 30° or 40°.

To show the real time application scenario, we perform the third experiment in a situation that the angle of $\theta_{\max} = 20^\circ$ and $\gamma_{\max} = 30^\circ$. The results show that the suitable setting of the initial numbers of probes can be around 3 to 5. We also observe that the system performs about 5.29 runs of perturbation process per second on average, run under the Red Hat Linux 7.2 system; the experimental machine

is equipped with Intel Pentium-4 CPU at 1G HZ. Detailed experimental results, including thousands lines of C source code implementations in UNIX system and many parameter settings, can be obtained through e-mail request to the corresponding author.

4 Future Work

Currently the system runs under the assumption that both sides of the paired proteins have the same number of atoms. In the future, we will consider the problem of *local structure alignment*, where the problem is trying to find the functional (or active) part of a given query protein. Furthermore, since the structure comparison problem, like many scientific computation/simulation problem, is very time-consuming under cases of large structures and large number of paired structures, it is desirable to implement the system under grid-environment to increase the throughput of the system.

References

[1] 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 Res.*, 28:235–242, 2000.

[2] J. M. Bujnicki. Phylogeny of the restriction endonuclease-like superfamily inferred from comparison of protein structures. *J Mol Evol.*, 50:38–44, 2000.

[3] C. Chothia and A. M. Lesk. The relation between the divergence of sequence and structure in proteins. *EMBO J.*, 5:823–826, 1986.

[4] S. Cristobal, A. Zemla, D. Fischer, L. Rychlewski, and A. Elofsson. A study of quality measures for protein threading models. *BMC Bioinformatics*, 2:5, 2001.

[5] S. Dietmann and L. Holm. Identification of homology in protein structure classification. *Nature Struct. Biol.*, 8:953–957, 2001.

[6] Z. Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*, 18:1:23–38, 1986.

[7] J. F. Gibrat, T. Madej, and S. H. Bryant. Surprising similarities in structure comparison. *Curr Opin Struct Biol.*, 6:377–385, 1996.

[8] L. Holm and C. Sander. Touring protein fold space with DALI/FSSP. *Nucleic Acids Res.*, 26:316–319, 1998.

[9] M. S. Johnson, M. J. Sutcliffe, and T. L. Blundell. Molecular anatomy: Phyletic relationships derived from three-dimensional structures of proteins. *J Mol Evol.*, 30:43–59, 1990.

[10] Yaw-Ling Lin, Ying-Hung Lin, Po-Shun Yu, and Hsun-Chang Chang. Randomized algorithms for three dimensional protein structures alignment. *The 6th International Symposium on Computational Biology and Genome Informatics.*, pages 122 – 125, 2005.

[11] A.C.R. Martin. `http://www.bioinf.org.uk/software/profit/`.

[12] A.D. McLachlan. Rapid comparison of protein structres. *Acta Cryst*, A38:871–873, 1982.

[13] K. Mehlhorn and St. Naher. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[14] J. T. Schwartz and M. Sharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *Int. J. Robotics Research*, 6:29–44, 1987.

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