Comparisons of Semi-local Alignment Algorithms for Protein Structures

Hong-Shin Chen Yaw-Ling Lin*

Department of Computer Science and Information Engineering,
Providence University,
200 Chung Chi Road, Shalu, Taichung County,Taiwan 433.
hongxin0118@gmail.com, yllin@pu.edu.tw

Abstract

Protein structures are essential for correct biological functions because similar structures between proteins allow molecular recognition. Identifying similar structures between proteins provide the opportunity to recognize homology that is undetectable by sequence comparison. Thus comparison and alignment of protein structures represents a powerful means of discovering functions, yielding direct insight into the molecular mechanisms. Currently, there are several techniques available in attempting to find the optimal alignment of shared structural motifs between two proteins.

In this paper, we improve segment strategy of our segment alignment. Secondly, we propose algorithms and develop tools for local alignment between two protein structures by the technique of two methods and semi-local alignment between the carbon atoms of two proteins.

These initialized alignment algorithms are later refined by our previously proposed methods by stepwise finding better alignments of the protein pairings using minimum bipartite matching method on geometric distance space and several other adjustment strategies in estimating good isometric transformations based on trigonometric series approximation.

We show the effectiveness of the proposed semi-local alignment algorithm by a set of experiments, which improve several previous results. Furthermore, some of our preliminary result is accessible through the web interface.

Keywords: *structural proteomics, algorithms, structure alignments and comparisons,* rmsd*, minimum bipartite matching, secondary structure, semi-local alignment.*

*Corresponding author. This work is supported in part by the National Science Council(NSC 98-2221-E-126-007), Taiwan, Republic of China.

1 Introduction

Protein structures play critical roles in vital biological functions [7]. With more than 40,000 protein structures determined by the advances in X-ray crystallography and NMR spectroscopy to date, molecular biologists these days proceed in the direction of analyzing and classifying these protein structures in order to discover the structural relationships with protein functions [5].

Detection of proteins with a similar fold can suggest a common ancestor, and often a similar function [4]. 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 [3, 11]. The link between structural classification and sequence families enables us to study functions of various folds, or whole proteins [13].

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 Vector Alignment Search Tool (VAST) system [8] is based on continuous distribution of domains in the fold space. The FSSP/DALI system [10] 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 [19] 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). 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 structure alignment computation. The first atom of the first selection is compared to the first atom of the second selection, fifth to fifth, and so on. Incorporating with ideas of bipartite matching and 3-parameter isometric transformation, Lin *et al.* [12, 17] proposed methods of using parametric searching strategies with adaptive controls, and demonstrated that more accurate and similar protein structure pairings are possible comparing to previous known results like VAST [8] or CE [19].

One of the crucial steps of these algorithms is finding a good isometric transformation, which leads to the best atom-pairing alignment between two proteins. In this paper, we propose algorithms of for efficiently locating more suitable isometric transformations of one structure and aligning it to the other structure. Based upon the periodical property of the parametric settings, we propose parametric searching strategies by approximations with power series and trigonometric series. We show the effectiveness of the proposed parametric searching strategies by a set of experiments, which leads to better alignments of structure pairing in general.

2 Background and Terminology

The main idea of our local refinement algorithm for finding a suitable matching between two sets of points before utilizing the RMSD procedure to fine-tune the final result is by adjusting the suitable parameter sets by ways of searching the underlying parametric space.

Root mean squared deviation

The smallest *root mean squared deviation (rmsd)* is a least-squares fitting method for two sequences of points [10]. 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

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

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 $d(P, Q) = \min_{R,\mathbf{a}} d(P, Q, R, \mathbf{a})$. We refer to Martin's ProFit package (standing for protein fitting system) [14] and write a program to calculate the *rmsd* between C-α atoms of paired protein backbones with C language. Fitting was performed using the McLachlan algorithm [15].

Isometric rotation transformation

According to Euler's rotation theorem [6], any rotation about the origin point can be described by using three angular parameters. The rotation is determined by 3 consecutive rotations with 3 *Euler angles* (α, β, γ). The first rotation is done by the angle α around the z-axis, the second is done by the angle β around the x-axis, and the third rotation is done by the angle γ around the z-axis. See [9] for more detailed discussions about the transformation.

Analog to Euler's setting, we can also consider the point of north-pole $\mathbf{n} = (0, 0, 1)^T$ on the unit sphere. After the rotation, R, say $\mathbf{n}$ is rotated to another point $\mathbf{p} = (x, y, z)^T$; 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 $\mathbf{p} = (x, y, z)^T$ on the surface of the unit sphere a *probe*. Note that the *movement* of each probe is started from the north-pole $(0, 0, 1)^T$ to other points in the sphere. The position of $\mathbf{p}p$ is de-

cided by the parameters (α, β), and exact rotation is fixed by the self-rotation angle γ.

As a result, we reduce the problem of finding the good rotation matrix to the new problem of finding a good 3-parameter setting. The rotation matrix is thus characterized by just adjusting the 3 uniformly distributed parameters.

The 3-parameter Rotation Matrix

Similar to Euler's rotation transformation, our 3-parameter method (α, β, γ) can be summarized as the following:

- **Rotation around z-axis:**

  Given a unit vector $\mathbf{p} = (x, y, z)^T$, $\mathbf{p}$ is transformed into $\mathbf{p}'$ by a rotation around the z-axis by angle α. That is, let

$$\mathbf{p}' = \begin{pmatrix} x_\alpha \\ y_\alpha \\ z_\alpha \end{pmatrix} = \begin{bmatrix} \cos\alpha\pi & \sin\alpha\pi & 0 \\ -\sin\alpha\pi & \cos\alpha\pi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{p}$$

  Since $\sin\theta = \alpha$ and thus, $\cos\theta = \sqrt{1-\alpha^2}$.

- **Rotation around x-axis:**

  The vector, $\mathbf{p}' = (x_\alpha, y_\alpha, z_\alpha)^T$, is transformed into the probe $\mathbf{p}''$ by a rotation around the x-axis by angle β. That is, let

$$\mathbf{p}'' = \begin{pmatrix} x_\beta \\ y_\beta \\ z_\beta \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta\pi & -\sin\beta\pi \\ 0 & \sin\beta\pi & \cos\beta\pi \end{bmatrix} \cdot \mathbf{p}'$$

  then we will get new coordinate of $(x_\beta, y_\beta, z_\beta)^T$.

- **Rotation around the probe $\mathbf{p}''$:**

  The last rotation matrix, R_γ, do the body rotation around the probe $\mathbf{p}''$ by angle γ; see [9] for related discussions about the transformation. That is, let

  $(x, y, z) = (x_\beta, y_\beta, z_\beta)^T$.
  $c = \cos\gamma\pi, s = \sin\gamma\pi, h = 1 - c$.

$$R_\gamma = \begin{bmatrix} c + x^2h & xyh - zs & xzh + ys \\ xyh + zs & c + y^2h & yzh - xs \\ xzh - ys & yzh + xs & c + z^2h \end{bmatrix}$$

Minimum bipartite matching

We use the minimum bipartite matching algorithm to find the best matching between two sets of points to decide the best matching for the *rmsd* procedure. We adopted the Munkres [16, 2] algorithm. In order to improve the efficiency of computation, we rewrite the Munkres algorithm, which is implemented on a public available perl module, by C language.

2.1 Parametric adjustment with trigonometric series

In the trigonometric series estimation algorithm [18], the three parameters (rotational angles) are assumed to be independent. Three parameters are adjusted one by one, and the estimation function corresponding to the affected *rmsd* values is subsequently better estimated by increasing the curvature degree of the estimated function. The trigonometric series function is described as the following:

$$\begin{aligned} f(\theta) = C_1 &+ C_2\cos\pi\theta + C_3\sin\pi\theta \\ &+ C_4\cos 2\pi\theta + C_5\sin 2\pi\theta \\ &+ C_6\cos 3\pi\theta + C_7\sin 3\pi\theta + \ldots \\ &+ C_{2k}\cos k\pi\theta + C_{2k+1}\sin k\pi\theta. \end{aligned} \tag{1}$$

where $f(\theta)$ denote the corresponding *rmsd* value with respect of one of the three parameters, (α, β, γ). The k usually reflects the numbers of local maximal points in the approximated curve.

2.2 Initialization by segment alignment

Comparing to the more sophisticated methods like CE [19] or VAST [8], the main-vector initialization position [18] does have the advantage of saving valuable computation resources. Yet the initial orientation found by the main-vector method does not produce satisfactory final orientation even after the fine-tune procedures. The idea here is trying to find more suitable starting positions and still conserve enough computation time for later adjustment. Since the protein structure is just a chain sequence of atoms, we can subdivide the sequence and use the subsequence matching information to find the better alignment. Thus, the atom chains of a structure is divided into several (consecutive) segments.

Given a list of (consecutive) atoms obtained from the PDB file [1], one way of dividing protein chains of a structure is by using the secondary structural information of the given protein. That is, for the secondary structural partition method, the segments of structures is determined by partition the protein sequence by the secondary structural information of the given protein. Another

possible division scheme is obtained by slicing a fixed number of atoms of the given protein. Thus, for the fixed number partition method, the segments of structures is determined by partition the protein sequence by a fixed number of atoms of the given protein.

After the segments of structures is decided, the *segment alignment* uses the standard dynamic programming technique to obtain feasible pairings between segments by maintaining a suitable score table. The dynamic programming evaluation function is described as the following:

$$score(s, \lambda) = Ump \cdot \mid s \mid$$
$$score(\lambda, t) = Ump \cdot \mid t \mid$$
$$score(sx, ty) =$$
$$\min \begin{cases} \textsc{Rmsd}(L(s,t) \circ \textsc{Match}(x,y)) \cdot \ell \\ \quad + Ump \cdot (\mid sx \mid + \mid ty \mid -2\ell) \\ score(sx, t) + Ump \cdot \mid y \mid \\ score(s, ty) + Ump \cdot \mid x \mid \end{cases}$$

here λ denotes the empty list; s and t are two prefix segment lists. $L(s,t)$ is the alignment between segment lists s and y, and n_L denotes the number of atoms in L; $\ell = \mid L(s,t) \circ \textsc{Match}(x,y) \mid$.

The recurrence relation for evaluating the value *score* relies on three possible alignments between sx and ty. Here s and t are two prefix *segment lists*, and x and y are the two currently (last) considered segments. The first alignment, L is the pairing list from $L(s,t)$ merging with $\textsc{Match}(x,y)$ which stands for the match between segment x y. Since $\textsc{Rmsd}()$ returns the average precalculated *rmsd* value, the number is multiplied by the number of matched pairs ℓ. However, if one can not find any match for an atom, a given punishment constant, Ump, must be added to encourage most atom be aligned with some atoms on the other sequence. Another possibility is the case of $score(sx, t)$; in that case, the segment y is not able to match with segment on the other list. Thus we need to add in the punishment values for all atoms of the y segment. The case of $score(s, ty)$ is also treated similarly.

3 Methodology

In this section, first we introduce the motivation about why we want to use the semi-local alignment algorithm as an improved version of the initial setting orientation before the following refinement algorithm to find the better list between two proteins. Secondly, the idea of the semi-local alignment dynamic programming techniques are addressed in more details as the initial setting algorithm for two protein structures.

3.1 Motivation

In our previous works, the trigonometric series estimation perturbing method is used to further improve the given protein structure alignments. It is shown that our proposed method does improve the results given by VAST by about 11% improvement by the alignment refinement algorithms. It is interesting to note that the best initial alignment algorithm we found so far is the VAST algorithm against all other algorithms including CE, segment alignment, or the main-vector method. On the other hand, note that the problem of 3D protein structure alignment of two proteins, say PA and PB, is actually a form of *semi-local alignment* variants. In order to compare our results with VAST's, we observe that the segment alignment method usually performs worse than the results of VAST when the length of PA is considerable much shorter comparing to PB. Note that the original segment alignment algorithm is essentially a *global alignment* algorithm, while we insist that each atom of PA must be aligned to some atom of PB, but not vice versa.

That is, the alignment we are looking for is a global alignment for PA in the sense that each atom of PA is aligned; however, it is a local alignment for PB since the prefix and suffix of these unaligned atoms of PB do not count.

3.2 Method

There ideas are similar to the *segment alignment* described previously in the paper; they both use the standard dynamic programming technique and maintain two dimensional score table.

In these three methods most notations are described similarly in segment alignment. Note that U_h (U_v) denote the horizonal (vertical) unmatched penalty. The input is assumed as two segmented list A and B.

Segment Alignment

In our previous works, the *segment alignment* input protein structure segment of list are cut three atoms. In this paper, we improve the *segment alignment* input protein structure segment list to cut one atom.

M1

This idea is change horizonal (vertical) unmatched penalty of two dimensional score table of our *segment alignment algorithm*.

$$score(s,\lambda) = U_v \cdot \mid s \mid;$$
$$score(\lambda,t) = U_h \cdot \mid t \mid;$$
$$score(sx,ty) =$$
$$\min \begin{cases} \textsc{Rmsd}(L(s,t) \circ \textsc{Match}(x,y)) \cdot \ell \\ +U_v \cdot (\mid sx \mid -\ell) + U_h \cdot (\mid ty \mid -\ell), \\ score(sx,t) + U_h \cdot \mid y \mid, \\ score(s,ty) + U_v \cdot \mid x \mid \end{cases}$$

Output of $score(A,B) = \min_{t \in \text{prefix}(B)} score(A,t)$.

M2

This idea to improve when the length of A is considerable much shorter comparing to B. If B all of unmatch atoms were given penalty, such the penalty of the B will be too high. However, the B total unmatch penalty be not over the A atoms.

$$score(s,\lambda) = U_v \cdot \mid s \mid;$$
$$score(\lambda,t) = U_h \cdot \mid t \mid;$$
$$score(sx,ty) =$$
$$\min \begin{cases} \textsc{Rmsd}(L(s,t) \circ \textsc{Match}(x,y)) \cdot \ell \\ +U_v \cdot (\mid sx \mid -\ell) + U_h \cdot \min\{\mid ty \mid -\ell, P_{max}\}, \\ score(sx,t) + U_h \cdot \mid y \mid, \\ score(s,ty) + U_v \cdot \mid x \mid \end{cases}$$

Output of $score(A,B) = \min_{t \in \text{prefix}(B)} score(A,t)$

The P_{max} is maximum unmatch penalty.

Semi-Local Alignment Algorithm

This idea of our *semi-local alignment* algorithm, the dynamic programming evaluation function is slightly altered to give the semi-local minimum valued as described in the following:

$$score(s,\lambda) = U_v \cdot \mid s \mid;$$
$$score(\lambda,t) = 0;$$
$$score(sx,ty) =$$
$$\min \begin{cases} \textsc{Rmsd}(L(s,t) \circ \textsc{Match}(x,y)) \cdot \ell \\ +U_v \cdot (\mid sx \mid -\ell) + U_h \cdot (\mid ty \mid -\ell), \\ score(sx,t) + U_h \cdot \mid y \mid, \\ score(s,ty) + U_v \cdot \mid x \mid \end{cases}$$

Output of $score(A,B) = \min_{t \in \text{prefix}(B)} score(A,t)$.

The rationale behinds the semi-local recurrence relation for evaluating the value *score* is similar to the well known Smith-Waterman's *local alignment* algorithm [20]. Except for the facts that we are looking for is a global alignment for A since each atom of A is expected to be aligned; however, it is a local alignment for B since the prefix and suffix of these unaligned atoms of B can be omitted by the initial rule that $score(\lambda,t) = 0$ and the final output: $score(A,B) = \min_{t \in \text{prefix}(B)} score(A,t)$.

3.3 Parametric adjustment with trigonometric series

By incorporating with the two key concepts, parameterized-rotation as well as bipartite matching, the main algorithm can compare paired protein structures once given a reasonable good initial setting of the 3 parameters. The proposed generic parametric searching algorithm is showed in Figure 1. Since the best parametric settings can be very difficult to locate, our previous methods concentrate on using randomized perturbation method in searching sufficiently large number of parametric probes over the parameter spaces and let each probe searching its own proximity in a randomized greedy manner. Here in this paper we demonstrate that the underlying corresponding *rmsd* values associated with the parametric sets are related to each other in periodical and *continuous* manner; thus seeking reasonable approximation of the underlying *rmsd* values distributions is possible by some suitable mathematical models, especially by trigonometric series.

4 Experimental Results And Web System

In this section, we introduce the target of experimental data set first. Then we show the difference with dividing protein chains of a structure depends on the different secondary structures of the given protein. Finally, we show the Web enable user can use our system through the network.

4.1 Data Set

We choose the PDB for our experimental sample source, and we randomly pick 14,400 protein structures in the PDB database as our experimental subjects by the uniform distribution sampling out of totally 63,093 protein structures as of Feb, 2010. For each chosen protein structures we randomly choose one structure alignments listed on the database of VAST as the tested targets. We

```
SEMI-LOCAL-ALIG(A, B)      ▷Semi-Local Alignment algorithm
    Input : Two segment list of protein atoms, namely (A[1], A[2], ...), (B[1], B[2], ...).
    Global : U_v, U_h.
             The U_v,U_h are real numbers.
                 ▷ U_v is unmatched penalty of vertical.
                 ▷ U_h is unmatched penalty of horizonal.
 1  for i ← 0 to n_s                                   ▷ initiate the table
 2      do score[i, 0] ← U_v · lenAs[i]
 3  for j ← 0 to n_t
 4      do score[0, j] ← 0
 5  for i ← 1 to n_s
 6      do for j ← 1 to n_t
 7             do L ← L[i − 1, j − 1] ∘ MATCH(i, j)
 8                r ← RMSD(L)
 9                s ← r · n_L + U_v · (lenAs[i] − n_L) + U_h · (lenBs[j] − n_L)      ▷n_L = (n_{L_[i−1,j−1]} + n_{M_[i,j]})
10                if s ≤ score[i, j − 1] + U_v · lenAs[i] and s ≤ score[i − 1, j] + U_h · lenBs[j]
11                   then score[i, j] ← s; L[i, j] ← L
12                elseif score[i, j − 1] + U_h · lenBs[j] ≤ s and score[i, j − 1] + U_h · lenBs[j] ≤ score[i − 1, j] + U_v · lenAs[i]
13                   then score[i, j] ← score[i, j − 1] + U_h · lenBs[j]; L[i, j] ← L[i, j − 1]
14                else
15                         score[i, j] ← score[i − 1, j] + U_v · lenAs[i]; L[i, j] ← L[i − 1, j]
16 min ← score[n_s, n_t] ← (lenAs[n_s] − m_{L_[n_s,n_t]}) · 2 · U_v
17 for k ← n_t − 1 to 1
18     do score[n_s, k] ← (lenAs[n_s] − m_{L_[n_s,k]}) · 2 · U_v
19        if score[n_s, k] ≤ min
20           then min ← score[n_s, k]
21                f ← k
22 return L[n_s, f]                  ▷ return the best of pairing list.
```

Figure 1: The Semi-Local Alignment algorithm .

use the term, P, to stand for one of the 14,400 randomly picked protein structures, and we use Q to stand for one of the neighbors of each P. Note that P and Q include all un-aligned and aligned atoms. We use the term, PA, to stand for the aligned atoms of P by VAST, and we use PB to stand for one of the neighbors of each PA. Totally, there have 14,400 protein pairings can test by our previous experiment. In this paper we randomly pick 1,000 protein pairings from 14,400 protein pairings to test our experiment.

4.2 Web System

We make a web for user who is interesting in our protein structures comparison system at `http://bioinfo.cs.pu.edu.tw/pupsc.html`. The user of our web service usually provides two protein PDB IDs. We provide three initial method and VAST. And provide two adjust method. Our web service will search and obtain the corresponding PDB data from the Protein data bank database [1] and perform the desired protein structure alignment/comparison using the chosen set of algorithms. To avoid the time delay, our web service also provides user access keys for user to check the result later on after they submit the desired query. User can come back and check the result after operation finished by our local server.

5 Concluding Remarks

In this paper, we propose algorithms to improve the *rmsd* value between a protein structure pair by finding better alignment list. A set of experiments is designed to test the parameters; these adjusted parameters are set to perform the experiments over a thousand of protein pairs, which are uniformly random sampled from the PDB database. As the results shown that our methods improve the alignment computed by the VAST by an averaged improvement ratios about 11 %. The results demonstrate that the method of using 3D Euclidean distance minimum bipartite matching with trigonometric series estimated parametric searching scheme indeed improves existed known system like the VAST.

The experiments show that a good initial rotation is very important before the parametric adjustment. The idea of our semi-local alignment algorithm does improve the results given by the segment alignment initial setting method. It is shown that our semi-local alignment algorithm

The initial method method	The average *rmsd* of initial method	The average *rmsd* after trigono. adj.
VAST	1.9572	1.7329 (11.46%)
semi-local-Alig	1.8378	1.7697 (9.58%)
cut1	2.1031	1.7947 (8.30%)
M1	1.9025	1.8047 (7.79%)
cut3	3.4563	1.8526 (5.34%)
M2	2.0612	1.9033 (2.75%)
Main Vector	4.2697	2.5907 (-32.37%)

Table 1: The *rmsd* values of trigonometric series adjustment after initial alignment of VAST, Seg-Alig(cut3), Seg-Alig(cut1), Semi-local-Alig, M1, M2 and Main Vector. Notice the improved ratio (in %) comparing to VAST's original *rmsd*.

discovers the initial orientations of protein pairs with *rmsd* values outperforms the settings given by VAST. Further improvements and parametric adjustments of the semi-local alignment methods are still possible for future study.

Finally, 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 massive parallel machines cluster, e.g., the 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 Research*, 28:235–242, 2000.

[2] F. Bourgeois and J. C. Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. In *Communications of the ACM*, volume 14, pages 802 – 804, New York, NY, 1971. USA.

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

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

[5] N. Echols, D. Milburn, , and M. Gerstein. Molmovdb:analysis and visualization of conformational change and structural flexibility. *Nucleic Acids Res.*, 31:478V482, 2003.

[6] L. Euler. Formulae generales pro trandlatione quacunque corporum rigidorum. *Novi Acad. Sci. Petrop.*, 20:189–207, 1775.

[7] M. Gerstein, R. Jansen, T. Johnson, J. Tsai, and W. Krebs. Motions in a database framework: from structure to sequence. *Rigidity Theory and Applications*, pages 401–442 (ed. M F Thorpe and P M Duxbury, Kluwer Academic/Plenum Publishers), 1999.

[8] J. F. Gibrat, T. Madej, and S. H. Bryant. Surprising similarities in structure comparison. *Curr Opin Struct Biol*, 6(3):377–385, 1996 Jun.

[9] A. Gray. A treatise on gyrostatics and rotational motion. MacMillan,London, 1918.

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

[11] 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.

[12] Y. L. Lin and S. P. Huang. Tools and algorithms for refined comparison of protein structures. In *The 6th WSEAS International Conference on Microelectronics, Nanoelectronics, Optoelectronics (MINO '07)*, Istanbul, Turkey, 2007.

[13] Y. L. Lin, Y. H. Lin, P. S. Yu, and H. C. Chang. Randomized algorithms for three dimensional protein structures alignment. *The 6th International Symposium on Computational Biology and Genome Informatics.*, pages 122 – 125, 2005.

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

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

[16] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5:32–38, 1957.

[17] H. S. Shin, S. P. Huang, and Y. L. Lin. Parametric searching algorithms with adaptive strategy for three dimensional protein structures alignments. In *National Computer Symposium (NCS'2007)*, pages 144–154, Taichung, Taiwan, 2007.

[18] H. S. Shin, Y. L. Lin, and W. D. Jiang. Protein structures alignment algorithms by parametric searching with trigonometric series. In *Proceedings of the 25th Workshop on Combinatorial Mathematics and Computation Theory*, pages 44–54, Hsinchu, Taiwan, 2008.

[19] 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.

[20] M. S. Waterman and T. F. Smith. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.