Local Refinement Algorithms for Protein Structure Comparison and Alignment

Wei-De Jiang Hsiang-Sheng Shin Yaw-Ling Lin*

Department of Computer Science and Information Engineering,
Providence University,
200 Chung Chi Road, Shalu, Taichung County, Taiwan 433.
winderek@gmail.com, u8213028@yahoo.com.tw, yllin@pu.edu.tw

Abstract. *Proteins are an important class of biological macromolecules present in all biological organisms. Protein structures are essential for correct function because it allows molecular recognition. Thus protein structures provide the opportunity to recognize homology that is undetectable by sequence comparison, and they are powerful means of discovering functions, yielding direct insight of the molecular mechanisms. In this paper, we propose algorithms and develop tools for local alignment between two protein structures by means of local adjustments.We show the effectiveness of the proposed refinement methods and initialization algorithms by a set of experiments; the results show improvement comparing to several previous results.*

Keywords: structural proteomics, structure alignments and comparisons, local refinement, initial alignment

1 Introduction

Protein structures play critical roles in vital biological functions [9]. The three dimensional structure of proteins is highly conserved during evolution [4]. Proteins 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, 19].

With more than 50,000 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 [6]. This is why structural alignment of proteins increases our understanding of more distant evolutionary relationships [3, 13]. The link between structural classification and sequence families enables us to study functions of various folds, or whole proteins [15].

The VAST system [10] is based on continuous distribution of domains in the fold space. The FSSP/DALI system [12] 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 [24] 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`.

Incorporating with ideas of bipartite matching and 3-parameter isometric transformation, Lin *et al.* [14, 22] 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 [10] or CE [24].

In our previous work [23], we propose algorithms for efficiently locating more suitable isometric transformations of one structure and aligning it to the other structure. In this paper, we propose algorithms for local refinement and the new initialization method.

2 Background and Terminology

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

*Corresponding author. This work is supported in part by the National Science Council (NSC-96-2221-E-126-002), Taiwan, Republic of China.

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 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) 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 [8] to obtain the best matching of P' and Q'. By the matched pairing, we perturb and refine the final alignment to obtain a possible lower *rmsd*.

2.1 Root mean squared deviation

The smallest *root mean squared deviation (rmsd)* is a least-squares fitting method for two sequences of points [12]. 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 $d(P, Q) = \min_{R,\mathbf{a}} d(P, Q, R, \mathbf{a})$. Schwartz [21] showed that $d(P, Q, R, \mathbf{a})$ is minimized when $\mathbf{a} = 0$ and $R = (A^t A)^{\frac{1}{2}} A^{-1}$, where the matrix $A = (A_{ij})$ $i, j = 1, 2, 3$ is given by $A_{ij} = \sum_{k=1}^{n} (p_k)_i (q_k)_j$, where $A^{\frac{1}{2}} = B$ 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 [17].

We refer to Martin's ProFit package (standing for protein fitting system) [16] 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 [17].

2.2 Isometric Rotation Transformation

According to Euler's rotation theorem [7] , any rotation about the origin point can be described by using three angles. 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 [11] for related discussions about the transformation.

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

2.3 Minimum Bipartite Matching

We use the minimum bipartite matching to find the best matching between two sets of points to decide the best matching for the *rmsd* procedure. We adopted the Munkres [18, 2, 1, 20] algorithm. The public available implementation is written with Perl language. To improve the efficiency of computation, we implement the Munkres algorithm and write hundreds lines of C Codes.

2.4 Parametric Adjustment with Trigonometric Series

In our previous work [23], the trigonometric series estimation method, the three parameters are assumed to be independent. We adjust the three parameters one by one and increase the power of the estimated function. The trigonometric series function is described as the following:

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

, where the $f(\theta)$ denote the corresponding value of *rmsd* 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.

3 Methodology

In this section, first we introduce the motivation about why we want to use the local refinement algorithm to find the better list between two proteins. Secondly, we show the initial algorithm according to the structure of protein. The detail experimental result is showed in next section.

3.1 Motivation

In our previous work, the trigonometric series estimation method is used to find a better position in protein structure comparison. When comparing with the VAST, there are 15.89% improvement by our proposed method.It is appropriate to the local alignment algorithm in finding the better alignment. Therefore,

STRUC-MIR(P, Q, A) ▷ Structure Alignment with Mirror.

Input: (P, Q, A), where $P = \{p_1, p_2, \ldots, p_{n_P}\}$ and $Q = \{q_1, q_2, \ldots, q_{n_Q}\}$ are two set of 3D coordinates of points, and A is a initial alignment.

Output: (r, A), where r is a sufficiently low *rmsd*, and A is the new alignment

```
1  started ← true
2  repeat  improve ← true
3          repeat (P, Q') ← Φ_A(P, Q) ; r ← r_A        ▷ adjust atoms of Q to atoms of P
4                 A' ← MBM(P(A), Q') ; succ ← false     ▷ P(A) is the aligned atoms of P
5                 if r_A' < r then succ ← improve ← true; A ← A'; Q ← Q'
6          until not succ
7          if not improve and not started then exit
8          improve ← false ; started ← false
9          repeat (P, Q') ← Φ_A(P, Q) ; r ← r_A        ▷ adjust atoms of Q to atoms of P
10                A' ← MBM(Q'(A), P) ; succ ← false    ▷ Q'(A) is the aligned atoms of Q'
11                if r_A' < r then succ ← improve ← true; A ← A'; Q ← Q'
12         until not succ
13 until not improve
14 return (r, A)
```

MBM$(P(A), Q')$ returns the minimum bipartite matching of two point sets $P(A)$ and Q'.

$\Phi_A(P, Q)$ ▷ adjust atoms of Q to atoms of P by the alignment, A.

Input: (P, Q), where $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_m\}$ are two set of 3D coordinates of points.

Output: Q', where Q' is adjusted from Q.

```
1  M_R ← ROT(A)        ▷ M_R is a rotation matrix calculated from the alignment, A.
2  Q' ← TRANS(P, Q, M_R)        ▷ adjust atoms of Q to atoms of P by M_R.
3  return Q'
```

Figure 1: The mirroring method tries to find a better local alignment by reflection atoms of two structures.

we propose a local refinement algorithm, mirroring method, to have a better alignment. The procedure for all the algorithms is shown in Figure 3. PA and PB are two protein structures. We get the fixed numbers of aligned atom in PA by initial algorithm and then proceed with trigonometric series estimation method to adjust the parameters. The use of the mirroring method depends on global alignment or local alignment. Besides, we also develop two new initial methods, main vector and segment alignment, to substitute for the well-known methods, such as the VAST and CE. In the following we introduce the local refinement by mirroring method, then initial with main vector or segment alignment.

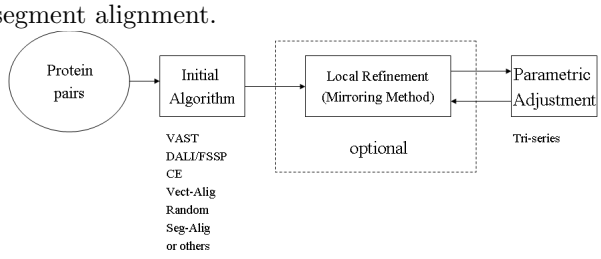

Figure 3: Algorithms for structure alignments of protein pairs.

3.2 Local Refinement by Mirroring Method

The principle for the mirroring method is to fix one side of protein pairings and find the minimum bipartite matching of the protein pairings. The mirroring algorithm is illustrated at Figure 1. Given P and Q two protein structures, let n_p and n_Q denote the numbers of atom in P and Q. Q' is the rotated Q. There is an initial alignment, A, whose length is n_A between P and Q, where $n_A \leq n_P, n_Q$. $P(A)$ stands for the aligned atoms in P, and $Q(A)$ stands for the aligned atoms in Q. $P(A)$ and $Q(A)$ are included in A. $\Phi_A(P, Q)$ means adjusting atoms of Q to atoms of P. The mirroring algorithm is divided into two parts:

1. Find a better matching by reflecting from $Q'(A)$ to P.
2. Find a better matching by reflecting from $P(A)$ to Q'.

The mirroring algorithm stops if it doesn't improve the presently best *rmsd* value for two consecutive times. The mirroring algorithm tries to find a better local alignment by reflection atoms of two structure. It fixes the numbers of atom for one side and finds a better matching in another side.

3.3 Initialization by Main Vector Method

The initial method, such as VAST and CE, supports the trigonometric series estimation method to improve the *rmsd* value. A better initial alignment is very important for the trigonometric series estimation method to adjust a better result. Therefore, we try to develop a initial method according to the shape of protein structure. The main vector method is to find a main vector about protein structure in 3-dimension and a second main vector in 2-dimension. We apply the inner and outer product to find the rotation and vertical vector. Let $\mathbf{x}$, $\mathbf{y}$ be two vectors and θ be the included angle of $\mathbf{x}$ and $\mathbf{y}$. We can have $\theta = \cos^{-1} \frac{\langle \mathbf{x} \cdot \mathbf{y} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$, then we use the

VECT-ALIG(P, deg, Z) ▷Vector Aligment ▷ Find the initial rotate for protein structure

Input: a set of 3D coordinates of points $P = \{p_1, p_2, \ldots, p_n\}$.
deg is the dimension for protein structure coordinates.
$(Z[2], Z[3])$ stands for (x-*axis*, z-*axis*).

Output: (P) rotated P

```
1  for i ← 3 to 2
2      do g ← CENTER(P, i, Z[i])      ▷ g is the mass center of P in the i-th dimension.
3         a ← FARTHEST(P, g)          ▷ a is a farthest point of P from g.
4         b ← FARTHEST(P, a)          ▷ b is a farthest point of P from a.
5         v ← b − a                   ▷ v is ab.
6         M_R ← MATRIX(V, Z[i])       ▷ M_R is the matrix which rotates V to Z[i].
7         P ← ROT(P, M_R)             ▷ rotate P by M_R.
8  return P
```

MATRIX(P, Q) returns the matrix that rotate point P to point Q on the unit sphere.

Figure 2: The initial rotation by the main vector method, When $i = 3$, there is a first main vector in 3-dimension and then rotate the protein structure by the matrix which rotates it to z-*axis*. When $i = 2$, there is a second main vector in 2-dimension which is x-*axis* and y-*axis*, and then rotate the protein structure by the matrix which rotates it to x-*axis*.

outer product to find the vertical vector, $\mathbf{v}$, which is defined as $\mathbf{v} = \mathbf{x} \times \mathbf{y}$, then we use θ and $\mathbf{v}$ to rotate the protein structure. The algorithm is shown in Figure 2. In this algorithm, we have a first main vector and a second main vector. If we assume **a, b** to stand for the two points of first main vector and **c, d** to to stand for another. There are totally four possible combinations for them, $(\overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{cd}}), (\overrightarrow{\mathbf{ba}}, \overrightarrow{\mathbf{cd}}), (\overrightarrow{\mathbf{ab}}, \overrightarrow{\mathbf{dc}}), (\overrightarrow{\mathbf{ba}}, \overrightarrow{\mathbf{dc}})$. We choose the minimum *rmsd* of them to be the initial rotation. Besides the main vector method, we also use a random initial rotation to execute the trigonometric series estimation method. The experimental results of those two different settings are discussed in next section.

3.4 Initialization by Segment Alignment

Comparing to the more sophisticated methods like CE or VAST, the main-vector initialization position does have the advantage of saving valuable processor computation resources. Yet the found initial orientation by the main-vector method seems a little bit rough and not being able to produces satisfactory final orientation even after the fine-tune procedures. The idea here is trying to find a more suitable starting position and still conserve enough computation time just for the better tryout. Since the protein structure is just a chain sequence of atoms, we can subdivide the sequence and use the subsequence matching information to find a better starting. Thus, the atom chains of a structure is divided into several (consecutive) segments. Here is a list of (consecutive) atoms appeared in the PDB file. One way of dividing protein chains of a structure depends on the secondary structures of the given protein. The other passable partitions can also be obtained by slicing a fixed number of atoms of the given protein. In the following experiment, we test the effectiveness of the method by using the fixed number partition method. 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:

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

here λ denotes the empty list; s, t are two 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 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, and the corresponding table lookup algorithm is shown in Figure 4. In this algorithm, we first initiate the table. $lenA[i]$ is prefix segment list when it treats the i-th segment. s is the score of new atoms just joint into. We choose the minimum and record into the score table. It also records $L[i,j]$ as the current merged list L. Finally, we get the value of the bottom right hand side corner in

```
SEG-ALIG   ▷Segment Alignment algorithm
 1  for i ← 0 to n_s                                   ▷ initiate the table
 2      do score[i, 0] ← Ump · lenAs[i]                ▷ Ump : unmatched penalty
 3  for j ← 0 to n_t
 4      do score[0, j] ← Ump · lenBs[j]
 5  for i ← 1 to n_s
 6  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 + Ump · (lenAs[i] + lenBs[j] − 2 · n_L)      ▷n_L ← (n_{L_[i−1,j−1]} + n_{M_[i,j]})
10         if s ≤ score[i, j − 1] + Ump · lenA[i] and s ≤ score[i − 1, j] + Ump · lenB[j]
11            then score[i, j] ← s; L[i, j] ← L
12         elseif score[i, j − 1] + Ump · lenB[j] ≤ s and score[i, j − 1] + Ump · lenB[j] ≤ score[i − 1, j] + Ump · lenA[i]
13            then score[i, j] ← score[i, j − 1] + Ump · lenB[j]; L[i, j] ← L[i, j − 1]
14         else
15                 score[i, j] ← score[i − 1, j] + Ump · lenA[i]; L[i, j] ← L[i − 1, j]
```

Figure 4: The segment alignment .

the table. And the $L[n_s, n_t]$ is the desired answer using the segment alignment method.

4 Experimental Results

In this section, we introduce the target of experimental data set first. Then we show the difference with VAST, CE, main vector, random and segment alignment. Finally, the experimental results for mirroring method is shown.

4.1 Data Set

We choose the PDB for our experimental sample source, and we randomly pick 200 protein structures in the PDB database as our experimental subjects by the uniform distribution sampling. For each chosen protein structures we randomly choose 30 structure alignments listed on the database of VAST as the tested targets. We use the term, P, to stand for one of the 200 randomly picked protein structures, and we use Q to stand for one of the 30 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. Totally, there are 6,000 protein pairings tested by our previous experiment.

4.2 Comparison of Five Methods

There are five different initial rotations in the following algorithms- VAST, CE, main vector, random and segment alignment. We use the numbers of atom found by VAST for our standard. CE always finds its own alignment, but we hope to compare the difference with the five methods in the same standard. Therefore, we also transform it back to the standard after the trigonometric series estimation method. When it finishes the trigonometric series estimation method, we use the PA which is standard from VAST to run one time MBM with Q. Therefore, the five methods are compared in the same PA. We randomly select 1,000 of 6,000 samples pairings for our experiments, and the distribution of them is shown in Figure 5

The MBM provides the effect of fine-tuning when there is a correct initial rotation. But there are still 2%, 2.3% and 4.9% to improve VAST, CE and segment alignment by trigonometric series estimation method. The result indicates that the initial alignment with a MBM is not perfect.

After VAST and CE execute the MBM and the trigonometric series estimation method, VAST drops the *rmsd* to 2.51. For this reason, it seems that VAST is a better initial seeding method than CE. The segment alignment drops the *rmsd* to 2.68 after execute the MBM and the trigonometric series estimation method. It is better then main vector, random and almost better then CE. It is very close VAST and sometimes defeat VEST. We think if we choose the way of dividing protein chains of the secondary structures, it should be more better.

4.3 Experimental Results for Mirroring method

We apply the mirroring method to the 6,000 protein pairings. We obtain 3.11 after the VAST, 2.55 after the VAST and a MBM and 2.27 after the VAST, a MBM and the mirroring method. For all the 6,000 sample pairings, the MBM improves the result of the VAST about 18.1%. The mirroring method improves the result of the VAST about 27%. The value of *rmsd* is down to 2.27 after mirroring method. We only need to execute 6.81 times MBM.

5 Concluding Remarks

In this paper, we develops algorithms to improve the *rmsd* value of a protein structure pair by finding better alignment of two structures. Our method substantially improves the alignments found by VAST method (the averaged improvement ratios is about 27%). A set of experiments is tested which leads to the conclusion

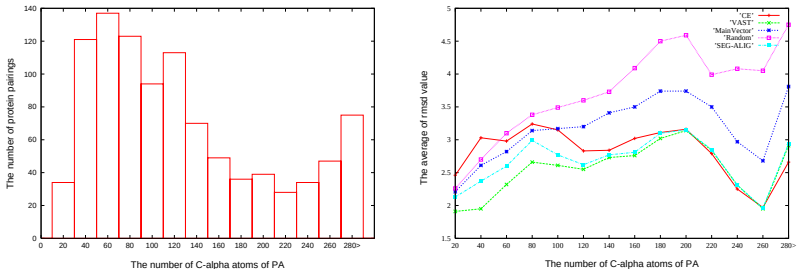

Figure 5: (a) The distribution of the 1000 randomly picked protein structure pairings. (b) The average of rmsd value for VAST, CE, Main Vector, Random and Segment alignment.

that good initialization orientation and its corresponding alignment list is crucial before adjusting parameters. Ways of finding suitable and feasible initalization orientation, including the VECT-ALIG and SEG-ALIGN methods, are proposed and tested; it can be concluded that the segment alignment method is a reasonable way of setting up the initial orientation of the given protein pair. Furthermore, the local refinement algorithm, the mirroring method, is proposed and the experimental results confirm the *rmsd* values can then be reduced substantially by the mirroring method.

References

[1] F. Bourgeois and J. C. Lassalle. Algorithm 415: Algorithm for the assignment problem (rectangular matrices). In *Communications of the ACM*, volume 14, pages 805 – 806, New York, NY, 1971. USA.

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

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

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

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

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

[9] 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.

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

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

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

[13] 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.

[14] 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.

[15] 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.

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

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

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

[19] C. A. Orengo, D. T. Jones, and J. M. Thornton. Protein superfamilies and domain superfolds. *Nature*, 372:631–634, 1994.

[20] R. A. Pilgrim. `http://csclab.murraystate.edu/-bob.pilgrim/445/munkres.html`.

[21] 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.

[22] 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.

[23] 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.

[24] 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.