Journal of Bioinformatics and Computational Biology
Vol. 4, No. 6 (2006) 1197–1216

ICP Imperial College Press
www.icpress.co.uk

MatAlign: PRECISE PROTEIN STRUCTURE COMPARISON BY MATRIX ALIGNMENT

ZEYAR AUNG*

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
azeyar@i2r.a-star.edu.sg

School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543
zeyaraun@comp.nus.edu.sg

KIAN-LEE TAN

School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543
tankl@comp.nus.edu.sg

Received 7 February 2006
Revised 4 August 2006
Accepted 5 August 2006

We propose a detailed protein structure alignment method named "MatAlign". It is a two-step algorithm. Firstly, we represent 3D protein structures as 2D distance matrices, and align these matrices by means of dynamic programming in order to find the initially aligned residue pairs. Secondly, we refine the initial alignment iteratively into the optimal one according to an objective scoring function. We compare our method against DALI and CE, which are among the most accurate and the most widely used of the existing structural comparison tools. On the benchmark set of 68 protein structure pairs by Fischer *et al.*, MatAlign provides better alignment results, according to four different criteria, than both DALI and CE in a majority of cases. MatAlign also performs as well in structural database search as DALI does, and much better than CE does. MatAlign is about two to three times faster than DALI, and has about the same speed as CE. The software and the supplementary information for this paper are available at http://xena1.ddns.comp.nus.edu.sg/∼genesis/MatAlign/.

Keywords: Protein structure; structural alignment; alignment quality criteria; structural classification.

1. Introduction

Comparison and/or alignment of three-dimensional (3D) protein structures plays a central role in structural bioinformatics. It is widely accepted that the function

*Corresponding author.

of a protein is more closely related to its 3D structure rather than its amino acid (AA) residue sequence. Structurally similar proteins perform similar functions. The comparative study of protein structures can give us knowledge of their functional relationships, and such knowledge can be applied in many applications such as drug design.

The problem of protein structure comparison has been studied since 1970s.[1] Nowadays, there are several structural comparison methods available: namely DALI,[2] CE,[3] SSAP,[4] Geometric Hashing,[5] VAST,[6] STRUCTAL,[7] StrAlign[8] and many more. Reviews on these methods can be found in survey papers and books by Koehl,[9] Novotny *et al.*[10] and Orengo *et al.*[1]

In this paper, we propose a new protein structure comparison method based on the alignment of distance matrices. It can provide precise results, and can be used for the detailed comparative structural analysis of proteins.

2. Preliminaries

In this section, we will discuss general information on protein structure comparison, distance matrix representation and alignment.

2.1. *Structural comparison framework*

When comparing two proteins, people usually try to find the corresponding pairs (i.e. alignment) of AA residues that provides the optimum similarity score with respect to the scoring scheme used. Thus, the terms "structural comparison" and "structural alignment" are often used interchangeably. (Still, there exist some non-alignment-based structural comparison methods.[1,11])

There are several ways to measure the similarity between two protein structures. Among them, *root mean square deviation* (RMSD) is the most commonly used.[9] Under this scheme, the aligned residues in one structure are superimposed onto those of another structure so as to yield the minimum RMSD value. The superimposition process involves translation and rotation of one structure with respect to the other. Mathematically, given two set of aligned residues A_{al} and B_{al} from two proteins A and B respectively, we have to minimize the RMSD value $\Delta(A_{\text{al}}, B_{\text{al}})$ between them:

$$\Delta(A_{\text{al}}, B_{\text{al}}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(A_{\text{al}}[i] - (\mathbf{R}\cdot B_{\text{al}}[i] + \mathbf{T}))^2} \tag{1}$$

where N is the number of aligned residues (i.e. $|A_{\text{al}}| = |B_{\text{al}}| = N$), and $\mathbf{R}$ and $\mathbf{T}$ are the rotation matrix and translation vector applied on B_{al} in order to yield the minimum RMSD value.

In most cases, RMSD alone cannot be used to determine the quality of an alignment. A smaller RMSD value does not always imply a better alignment quality. The length of alignment (i.e. the number aligned residue pairs) is also needed to

be considered. Suppose we have two structural comparison methods for aligning two protein structures with 100 residues each. If the first method can produce 30 aligned residue pairs with RMSD value 2.0 Å, and the second method can generate 60 pairs with RMSD 2.1 Å, the latter can be considered more significant.

Different groups of researchers have proposed different scoring schemes or functions to balance the RMSD value (Δ) and the number of aligned residue pairs (N). Each scheme calculates a "single-value" score by manipulating Δ and N in some manner so as to measure the quality of an alignment in its own way. There is no universal consensus on measuring the alignment quality by a single value.[9]

Here, we will look at the following four scoring schemes. The first one is used as the native scoring function for our proposed MatAlign method. The other three have been recently used as the quality criteria for comparative performance evaluation of different alignment methods.[12]

(1) Alexandrov and Fischer's *alignment score*[13] (denoted as S here). An alignment with the larger S value is considered to be a better one (i.e. S is to be maximized).

$$S = \frac{3 \times N}{1 + \Delta} \tag{2}$$

(2) Kleywegt and Jones' *similarity index* (SI)[14] (to be minimized).

$$\mathrm{SI} = \frac{\Delta \times \min(|A|, |B|)}{N} \tag{3}$$

where $|A|$ and $|B|$ are the lengths or the number of residues in the original proteins A and B, respectively.

(3) Kleywegt and Jones' *match index* (MI)[14] (to be maximized).

$$\mathrm{MI} = \frac{1 + N}{(1 + \Delta/w_0) \times (1 + \min(|A|, |B|))} \tag{4}$$

where we use $w_0 = 1.5$ as a default value.[12]

(4) Subbiah *et al.*'s *structural alignment score* (SAS)[15] (to be minimized).

$$\mathrm{SAS} = \frac{\Delta \times 100}{N} \tag{5}$$

2.2. *Distance matrix representation*

PDB (Protein Database Bank) (http://www.rcsb.org/pdb) provides the 3D (x, y, z) coordinates of constituent atoms of AA residues in each protein structure. When comparing the protein structures, people usually do not take all the atoms in the structure, but only the C_α (central carbon) atoms of AA residues into account.[2]

A 3D protein structure can be represented as a 2D distance matrix. The distance matrix $\mathcal{DM}_A$ of protein A with $|A|$ residues is an $|A| \times |A|$ matrix storing the pairwise inter-atomic distances among the C_α atoms in the protein. The distance

between two C_α atoms i and j ($1 \leq i, j \leq |A|$) is denoted as d_{ij}. Obviously, $d_{ij} = 0$ if $i = j$.

Representing a protein structure as a distance matrix is useful because it is rotation and translation invariant, yet it still can capture all the structural information about a protein as much as the original 3D representation can.

In the distance matrix of protein A, the ith row (denoted as $\mathcal{DM}_A[i]$) can be regarded as the *distance profile* of residue i, because it stores the C_α–C_α distances of residue i with respect to the other residues in A. For example, the first row of the distance matrix is the distance profile of residue #1.

2.3. *Aligning distance matrices for structural alignment*

In order to structurally align two proteins, we can align their distance matrices instead of their original 3D structures.[2] Alignment of distance matrices is based on the fact that two structurally matched residues, one from each protein, have the similar distance profiles (represented as rows in their respective distance matrices).

For example, suppose we have two protein structures P and Q which are identical except for the inserted residue 3 in P, as shown in Fig. 1. For simplicity, let us denote the distances between residues as symbols: A for d^P_{12} (C_α–C_α distance between residue 1 and 2 in P), B for d^P_{13}, C for d^P_{14}, etc. Since the two proteins are identical except for one residue, their corresponding C_α–C_α distances are the same; i.e. $d^P_{12} = d^Q_{ab} = A$; $d^P_{14} = d^Q_{ac} = C$, etc.

When we observe the distance matrices $\mathcal{DM}_P$ and $\mathcal{DM}_Q$ of proteins P and Q respectively, we can see that row $\mathcal{DM}_P[1]$ (i.e. the distance profile of residue 1 in protein P) is more similar to row $\mathcal{DM}_Q[a]$ (the distance profile of residue a) than any other rows in $\mathcal{DM}_Q$. The row–row matching score of $\mathcal{DM}_P[1]$ and $\mathcal{DM}_Q[a]$ is 4. The pairs (0–0), (A–A), (C–C) and (D–D) are the matching ones. The row–row matching score of $\mathcal{DM}_P[1]$ and any other row in $\mathcal{DM}_Q$ is at most 1. For example, if we take $\mathcal{DM}_Q[b]$, there can be only one match: either (0–0) or (A–A). (Both

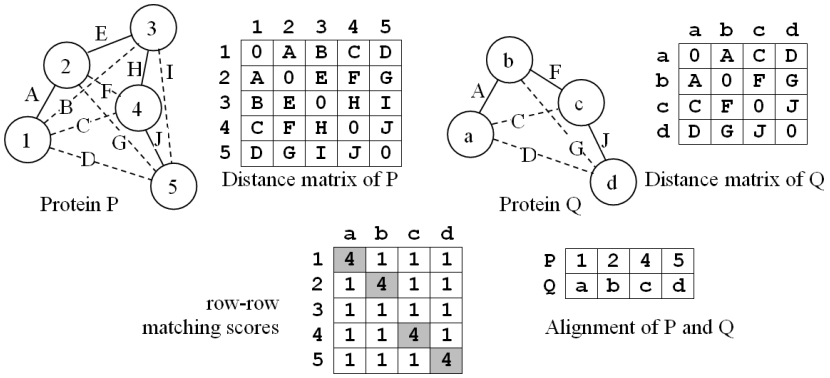


Fig. 1. Alignment of distance matrices.

matches cannot be achieved at the same time, because of their different sequence orders: $0, A$ and $A, 0$.) Similarly, we can observe that $\mathcal{DM}_P[2]$ is most similar to $\mathcal{DM}_Q[b]$; $\mathcal{DM}_P[4]$ to $\mathcal{DM}_Q[c]$; and $\mathcal{DM}_P[5]$ to $\mathcal{DM}_Q[d]$ as shown in Fig. 1. Thus, we finally have the alignment of residue pairs: $(1-a), (2-b), (4-c)$ and $(5-d)$.

3. The MatAlign Method

We propose a protein structure comparison method in the conventional framework of structural alignment, RMSD, and alignment score, using the principles of distance matrix representation and alignment as described above. We name our method **MatAlign** which stands for **Mat**rix **Align**ment. From the experimental results, it is observed that MatAlign can offer the precise alignment results. It is ideal for the detailed comparative analysis of protein structures.

The idea of aligning distance matrices to yield the alignment of protein structures has been previously used in the DALI[2] method. However, MatAlign adopts a different approach. DALI sub-divides a distance matrix into 6×6 overlapping sub-matrices, finds the matching sub-matrix pairs from two proteins, and assemble these matching pairs into the final alignment by means of Monte Carlo optimization. On the other hand, MatAlign uses dynamic programming at two levels: first for row–row alignment and second for consolidating row–row scores into the initial alignment; and then iteratively refining the initial alignment into the final one based on the objective alignment score function.

Again, although MatAlign utilizes the two-level dynamic programming strategy, it is substantially different from the double dynamic programming of SSAP.[4] The two methods are diverse in their data representation, superimposition and score accumulation strategies. (See the supplementary webpage for details.)

In addition, unlike DALI and many other methods such as VAST,[6] MatAlign does not use any secondary structure information at all. Thus, the alignment results produced by MatAlign will not be affected by the choice of the secondary structure annotation method.

MatAlign can be easily parallelized. Most of the running time of MatAlign is incurred in the step of all-against-all alignments of rows from two matrices. Since we have to perform multiple mutually-independent dynamic programming procedures in this step, we can simply reduce the running time by parallelizing them.

The basic MatAlign algorithm works in two steps. First, it finds the initial alignment of residues. Second, it refines the initial alignment into the final one with the optimum alignment score. We also implement some enhancements on the basic algorithm.

3.1. *Step 1: finding initial alignment*

The algorithm for generating the initial alignment between two protein structures A and B is described in Fig. 2.

```
function GetInitAlignment (DM_A, DM_B)
input:  (1) DM_A[1...|A|, 1...|A|] (distance matrix of protein A)
        (2) DM_B[1...|B|, 1...|B|] (distance matrix of protein B)
output: (1) N (number of aligned residue pairs)
        (2) A_al[1...N] (residues from A that involve in alignment)
        (3) B_al[1...N] (residues from B that involve in alignment)
procedure:
1.  for i = 1 to |A|
2.      for j = 1 to |B|
3.          SM[i,j] = AlignRow (DM_A[i], DM_B[j])
            /* row–row matching score of i^th row of DM_A and j^th row of DM_B */
4.  GS = 0 /* Gap score */
5.  for i = 0 to |A|    F[i,0] = i × GS    /* F is dynamic programming's matrix */
6.  for i = 0 to |B|    F[0,i] = i × GS
7.  for i = 1 to |A|
8.      for j = 1 to |B|
9.          F[i,j] = max{ (F[i−1,j] + GS), (F[i−1,j−1] + SM[i,j]), (F[i−1,j] + GS) }
10. GetAlignedPair (F, A_al, B_al, N)
11. return (A_al, B_al, N)
```

Fig. 2. Initial alignment generation algorithm.

As discussed in Sec. 2.3, we first have to compare all rows (representing distance profiles of residues) from $\mathcal{DM}_A$ against all rows from $\mathcal{DM}_B$, and store the row–row matching scores in the score matrix $\mathcal{SM}$. Row–row comparison algorithm **AlignRow** (line 3 in Fig. 2) is an adaptation of the classical Needleman–Wunsch dynamic programming algorithm used for sequence alignment.[16] We use the linear gap penalty model with the default gap penalty value of 0. We use the function Match$(\bullet, \bullet)$ to determine the degree of match between two C_α–C_α distance values $d1$ and $d2$.

$$\text{Match}(d1, d2) = \begin{cases} \alpha/(|d1 - d2| + \alpha) & \text{if } |d1 - d2| \le T_{\text{Match}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where α is a score adjusting weight and T_{Match} is a difference threshold of the distances. We use the empirically chosen values $\alpha = 0.75$ and $T_{\text{Match}} = 1.6\,\text{Å}$ that result in best performance. The function Match$(\bullet, \bullet)$ is used in the dynamic programming's selection step. After executing the dynamic programming, we get the matching score of the two given rows.

Suppose we have two proteins structures A and B whose distance matrices $\mathcal{DM}_A$ and $\mathcal{DM}_B$ are as shown in Fig. 3. As an example, the alignment of row $\mathcal{DM}_A[1]$ (the first row of A's distance matrix) and row $\mathcal{DM}_B[1]$ (the first row of B's distance matrix) is shown in Fig. 4. The alignment path is shown in gray. The matching score of $\mathcal{DM}_A[1]$ and $\mathcal{DM}_B[1]$ is stored in cell $[1, 1]$ of the score matrix $\mathcal{SM}$. In this manner, we align every row from $\mathcal{DM}_A$ and every row from $\mathcal{DM}_B$

	1	2	3	4	5	6	7
1	0.00	11.00	1.00	2.00	3.00	4.00	16.00
2	11.00	0.00	12.00	13.00	14.00	15.00	17.00
3	1.00	12.00	0.00	5.00	6.00	7.00	18.00
4	2.00	13.00	5.00	0.00	8.00	9.00	19.00
5	3.00	14.00	6.00	8.00	0.00	10.00	20.00
6	4.00	15.00	7.00	9.00	10.00	0.00	21.00
7	16.00	17.00	18.00	19.00	20.00	21.00	0.00

Distance Matrix of Protein *A*

	1	2	3	4	5	6
1	0.00	1.05	2.10	3.15	11.05	4.20
2	1.05	0.00	5.05	6.10	12.10	7.15
3	2.10	5.05	0.00	8.20	13.15	9.05
4	3.15	6.10	8.20	0.00	14.20	10.10
5	11.05	12.10	13.15	14.20	0.00	15.05
6	4.20	7.15	9.05	10.10	15.05	0.00

Distance Matrix of Protein *B*

Fig. 3. Two sample distance matrices of proteins A and B.

all-against-all, and fill their respective matching scores in the score matrix $\mathcal{SM}$ as shown in Fig. 5 (left).

Then, we apply another Needleman–Wunsch style dynamic programming algorithm on $\mathcal{SM}$ to generate the initially aligned residue pairs. In fact, the score matrix stores the degrees of matching of A's residues to B's residues, and dynamic programming effectively solves the *ordered bipartite matching* problem of maximizing the total degree of residue–residue matchings.

Figure 5 (right) shows the dynamic programming's matrix on $\mathcal{SM}$ and the initial alignment of A and B. In this alignment, we can observe that the residues whose matching partners cannot be successfully found are aligned with the gaps. Such a residue usually have a distance profile which is quite different from the others' in its counterpart protein. (The distance profiles of such residues are highlighted in gray in Fig. 3.) Since we use the ordered bipartite matching strategy, even though the distance profile of residue 2 in protein A is similar to that of residue 5 in protein B, they are not aligned together. This is because aligning $(2 - 5)$ will forbid the alignments of other good pairs $(3 - 2), (4 - 3)$ and $(5 - 4)$, and hence will result

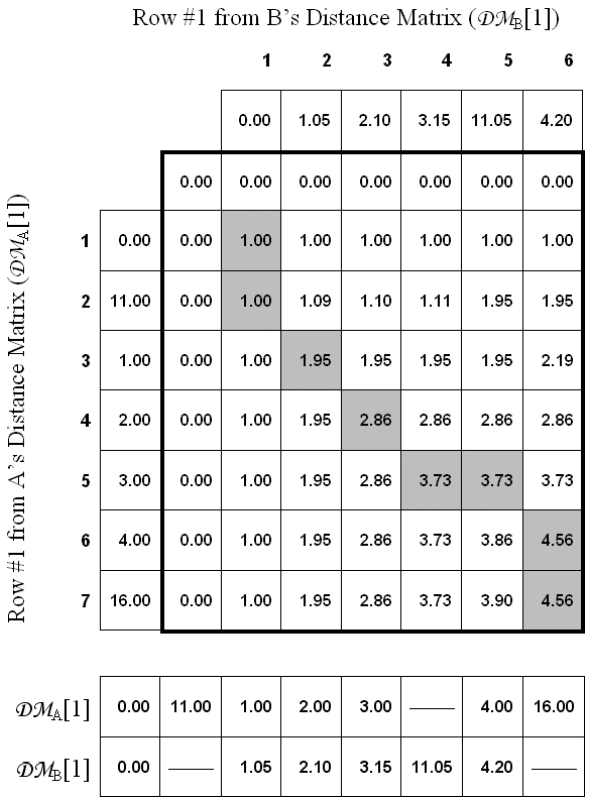


Fig. 4. Alignment of first row from distance matrix of A and that from B.

in a smaller total degree of residue–residue matchings. The actual aligned residue pairs are traced back from the dynamic programming's matrix F by a recursive algorithm **GetAlignedPair** (line 10 in Fig. 2) as described in the book by Setubal and Meidanis.[16]

3.2. *Step 2: refining alignment*

We use the alignment score S defined in Eq. (2) as MatAlign's native score. The function S balances the RMSD value and the number of aligned residue pairs.[13] Our objective is to maximize S as much as possible.

The initial alignment generated in Step 1 is usually not an optimal one in terms of S. Thus, we refine the alignment iteratively until S cannot be further improved. The refinement algorithm is given in Fig. 6.

In order to calculate the alignment score S, we first have to superimpose the set of aligned residues in one protein onto their counterparts in the other protein, and calculate the value of Δ (RMSD) between them. We can solve the RMSD problem

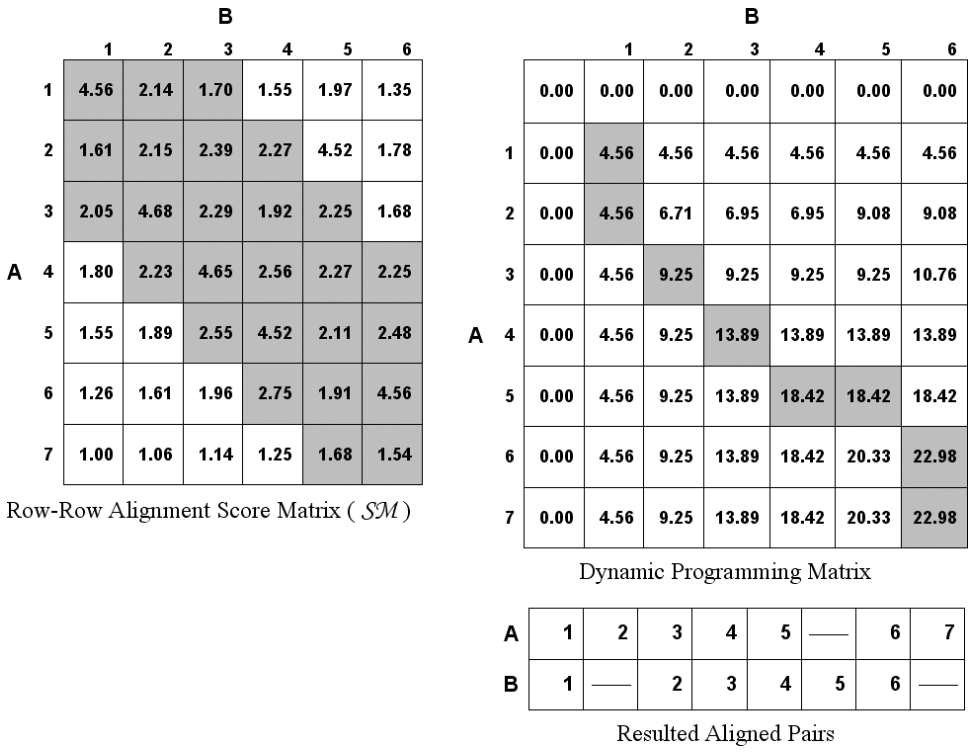


Fig. 5. Generating initial alignment of proteins A and B.

by using the *singular value decomposition* method. The function **CalculateRMSD** (line 3 in Fig. 6) is based on the one given in the work by Wu.[17]

During the process of calculating RMSD, we can easily pick up the pair of residues that are farthest. We remove this pair from our alignment, and iterate the processes of superimposition and calculating RMSD as long as the alignment score S converges (i.e. keeps increasing). We stop the iteration when S cannot be further improved (line 5 in Fig. 6).

The distribution of the RMSD values and the alignment lengths of 68 test protein pairs before and after the refinement step are depicted in Figs. 7 and 8, respectively. It can be observed that there are many alignments that are not well-fitted (i.e. large RMSD values) before the refinement. The number of such bad alignments is much reduced after the refinement.

3.3. *Enhancements on basic algorithm*

The following enhancements are done on top of the basic MatAlign algorithm in order to achieve better speed and greater accuracy.

function **RefineAlignment** $(N, A_{init}, B_{init}, N', A_{final}, B_{final}, \Delta)$
input: (1) N (number of initially aligned residue pairs)
(2) $A_{init}[1 \ldots N]$ (3D coordinates of residues from A that involve in initial alignment)
(3) $B_{init}[1 \ldots N]$ (3D coordinates of residues from B that involve in initial alignment)
output: (1) N' (number of finally aligned residue pairs)
(2) $A_{final}[1 \ldots N']$ (residues from A involving in final alignment)
(3) $B_{final}[1 \ldots N']$ (residues from B involving in final alignment)
(4) Δ (RMSD between A_{final} and B_{final})
procedure:
1. $S_{old} = 0;\ N' = N;\ A_{final} = A_{init};\ B_{final} = B_{init}$ /* Initialize */
2. **while** (TRUE)
3. $\Delta =$ **CalculateRMSD** $(N', A_{final}, B_{final})$ /* algo. given in Wu[17] */
4. $S = 3 \cdot N'/(1 + \Delta)$ /* Eq. 2 */
5. **if** $(S < S_{old})$ **then exit while** /* if the score diverges, then stop */
6. $S_{old} = S$ /* mark the last best score */
7. Remove the farthest residue pair from A_{final} and B_{final}
8. $N' = N' - 1$ /* reduce number of aligned pairs */
9. **return** $(N', A_{final}, B_{final}, \Delta)$

Fig. 6. Refining initial alignment into final alignment.

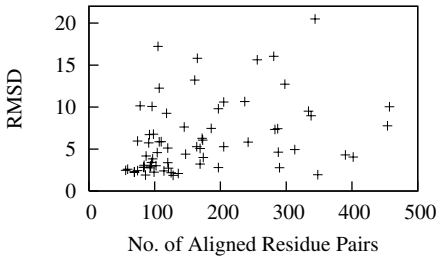


Fig. 7. Distribution of RMSD and alignment length before refinement.

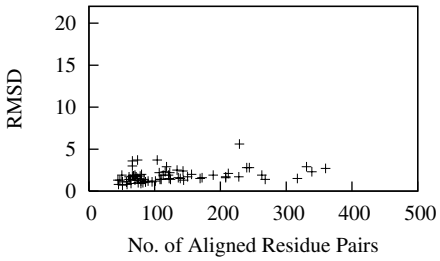


Fig. 8. Distribution of RMSD and alignment length after refinement.

- **Reduced rows**: In the row–row alignment step, it is observed that the large C_α–C_α distance values are not very important in determining the row–row matching scores, and hence can be ignored. These distance values are removed from the rows of the distance matrix, and the resultant reduced rows can

be used for alignment. For example, in Fig. 3, if we use the cutoff distance of 10 Å, the reduced version of the first row of A's distance matrix will be: $\{0.00, 1.00, 2.00, 3.00, 4.00\}$. In our actual implementation, we use the cutoff value 21 Å which is empirically determined.

- **Alignment within a band**: Both in the alignment of rows and alignment of the score matrix, it can be observed that the two residues whose ordinal positions are quite different rarely align. For example, in Figs. 3 and 4, when we align the first two rows of A and B, A's cell #2 and B's cell #5 are not likely to be aligned although their values are quite close. So, we define a *band*, and only the residue pairs that fall into the band are considered for alignment. In other words, for residue i and j to be aligned, the condition $(i - \text{Bandwidth} \leq j \leq i + \text{Bandwidth})$ must be satisfied. The Bandwidth can be calculated as:

$$\text{Bandwidth} = \#\text{Gaps} + \text{abs}(|A| - |B|) \tag{7}$$

  where #Gaps is the number of allowable gaps and $\text{abs}(|A| - |B|)$ is the length difference between A and B. In Fig. 5, the residue pairs falling into the bandwidth of 2 are shown as gray. In our implementation, we use $\#\text{Gaps} = 50$. The use of reduced rows and bands significantly improves the speed of the scheme as discussed later in Sec. 4.1.5.
- **Application of weights to row–row matching scores**: In the cases of distantly related protein structures, for a row in one protein's distance matrix, there are several rows in the other distance matrix which give the very similar row–row matching scores. As such, the initial alignment path based on these not-too-different scores may sometimes be incorrect. To reduce this effect, we multiply the row–row matching scores with the percent of the aligned residues. For example, in Fig. 5, $\mathcal{SM}[1, 1]$ will now be $4.56 \times (5/6)$, because the alignment of A's row #1 and B's row #1 results in five aligned residue pairs out of six pairs which is maximally possible. This heuristics improves the accuracy of the scheme.
- **Use of multiple initial alignment seeds**: Sometimes the default initial alignment produced from the first step may not lead to the optimal final alignment in the second step. To explore the possibilities for a better final alignment, we have to try multiple initial alignments. When extracting the initial alignment path from the dynamic programming's matrix, we set a threshold and if the value in a matrix's cell is lower than the threshold, we avoid this cell in our alignment path. We generate 100 different initial alignment paths using 100 different threshold values, refine each path, and select the one that gives us the best score S. This approach substantially improves the scheme's accuracy although it slightly affects the scheme's run-time efficiency. The combined effect of the use of row–row matching weights and the use of multiple alignment seeds on the accuracy is discussed later in Sec. 4.1.5.

3.4. *Time complexity*

The worst-case time complexity for finding the initial alignment of two proteins A and B with $|A|$ and $|B|$ residues, respectively, is $O(|A|^2|B|^2)$. (Every row in A's distance matrix has to be compared against every row in B's distance matrix, and each comparison using dynamic programming costs $O(|A||B|)$.) Nonetheless, because of the utilization of reduced rows and bands as mentioned in the above sub-section, the actual running time is reasonably fast.

The worst-case time complexity for the refinement step is $O(\min(|A|, |B|)^2)$. (The maximum possible length of the initial alignment is $\min(|A|, |B|)$, and thus at most $\min(|A|, |B|)$ refinement steps will be required. Each refinement step involves the calculation of RMSD which can be carried out in linear time, i.e., $O(\min(|A|, |B|))$ time.)

4. Experimental Results and Discussions

We assess the accuracy and speed performances of our proposed MatAlign method in relation with the established structural alignment methods, namely DALI[2] (DaliLite implementation[18] v2.4, released 2000) and CE[3] (latest version, released 2004). These two are among the most accurate and the most widely used of the existing structural alignment methods.[10] We conduct two types of experiments to access the respective accuracies of MatAlign, DALI and CE. Firstly, we measure the qualities of their alignment results using different criteria. Secondly, we evaluate the biological relevances of their alignment results by means of a database search test. We also compare the speeds of the three methods in both experiments.

Although MatAlign should also be compared with SSAP,[4] which also uses the two-level dynamic programming approach, we omitted this because the latter cannot be successfully ported to our computing platform. However, in terms of run-time efficiency, we can conjecture that MatAlign may be four to six times faster than SSAP. The experimental results by Kolodny *et al.*[12] show that SSAP is two times slower than DALI, and we show here that DALI in turn is two to three times slower than MatAlign. In terms of accuracy, Kolodny *et al.* reports that DALI provides better results (larger ROC area values) than SSAP in database search tests. Here, we show that MatAlign provides better results than DALI in alignment quality test and as good results as DALI in database search test.

4.1. *Experiment 1: on alignment quality*

Here, we assess the alignment quality of MatAlign in relation with those of DALI and CE. We use the benchmark of 68 protein pairs selected by Fischer *et al.*[19]

4.1.1. *RMSD and alignment length*

In Fig. 9, it can be observed that MatAlign generally tends to produce a smaller RMSD value (which means better fitted alignment) than DALI and CE do. The

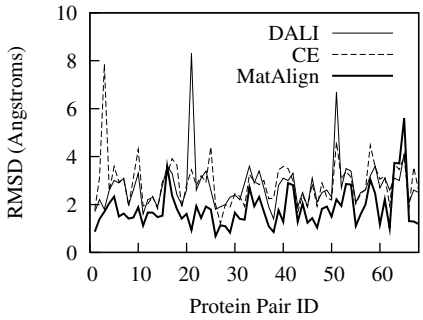


Fig. 9. Distribution of RMSD values.

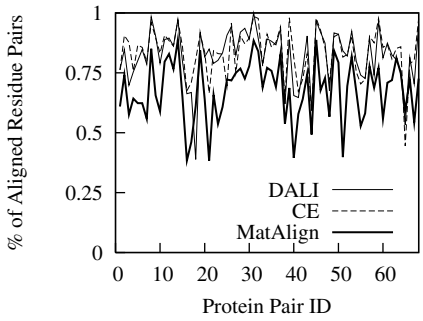


Fig. 10. Distribution of percents of aligned residue pairs.

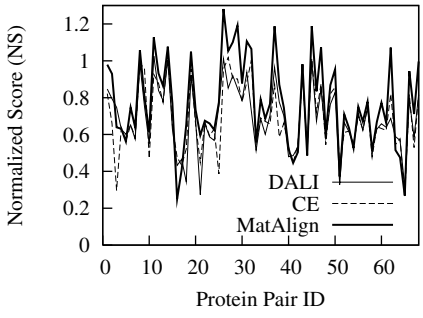


Fig. 11. Distribution of normalized score (NS) values. (Higher values mean better alignments.)

average RMSD of MatAlign for 68 benchmark protein pairs is 1.81 Å, and those of DALI and CE are 2.77 and 2.88 Å respectively.

On the other hand, MatAlign's alignment length is relatively shorter than those of DALI and CE as can be seen in Fig. 10. For simplicity of presentation in the figure, the alignment length is converted to the *percent of aligned residue pairs*,

which is a ratio of the number of aligned residue pairs to the length of the shorter protein. (The length of the shorter protein is the maximum possible length of the alignment.) The average percent of aligned residue pairs of MatAlign is 67%, and those of DALI and CE are 82% and 83%, respectively.

Nonetheless, MatAlign's alignment length is significantly large enough. It covers over at least 50% of the maximum possible alignment length in 91% (62 out of 68) of the cases, and at least 35% of the maximum length in all of the 68 cases. Thus, it can be concluded that MatAlign is able to detect the highly conserved yet significantly large structural cores in proteins.

4.1.2. *Accuracy assessment by different criteria*

It is observed that in terms of the alignment score criterion S by Alexandrov and Fischer[13] described in Eq. (2), MatAlign can achieve better (higher) alignment scores than DALI in 55 out of 67 cases[a] (i.e. 81%), and better scores than CE in 54 out of 67 cases[b] (79%). We show the distributions of the score values for DALI, CE and MatAlign in Fig. 11. Again, for convenience of presentation, the alignment score (S) is translated into the *normalized score* (NS), which is the ratio of S to the maximum possible alignment length. The average NS value of MatAlign is 0.77 whilst those of DALI and CE are 0.68 each.

However, this result may not be very convincing of the better accuracy achievement of MatAlign, because the scoring criterion S is also used as the native score of MatAlign. On the other hand, DALI and CE use their respective Z-scores as their native scores. This means that while the score of MatAlign is optimized in terms of S, those of DALI and CE are not.

Thus, we also compare MatAlign with DALI and CE using the other three scoring criteria which are not the native scores of any of these three methods. These are similarity index (SI)[14] (Eq. (3)), match index (MI)[14] (Eq. (4)), and structural alignment score (SAS)[15] (Eq. (5)). These three have also been used in a recent evaluation study on various structural alignment methods by Kolodny *et al.* (2005)[12].

It is observed that MatAlign can provide better results than both DALI and CE in a majority of cases in terms of all these three criteria! The summary of the accuracy comparison of MatAlign against DALI and CE in terms of all the four criteria is presented in Table 1. The detailed statistics are given in the supplementary information webpage.

4.1.3. *Accuracy assessment by adjusted RMSD*

In addition to comparing the alignment accuracies of DALI, CE and MatAlign in terms of the above four criteria, it will be interesting to compare these methods in

[a]DALI cannot produce any alignment result for `1mdc` vs `1ifc`.
[b]CE cannot produce any alignment result for `1bbt1` vs `2plv1`.

Table 1. Accuracy comparison of MatAlign vs. DALI and CE on the benchmark protein pairs by Fischer *et al.*

No. of cases (out of 67)	S	SI	MI	SAS
MatAlign is better than DALI	55 (81%)	58 (85%)	52 (76%)	58 (85%)
MatAlign is better than CE	54 (79%)	60 (88%)	53 (78%)	60 (88%)

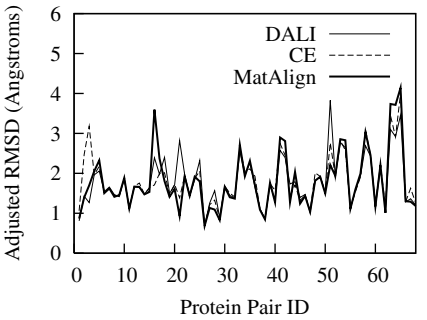


Fig. 12. Distribution of adjusted RMSD values. (Curve smoothing is used for the missing values.)

terms of their "adjusted" RMSD values at a fixed alignment length. For a given alignment case, from the three different alignment results by the three methods, we choose the one with the shortest alignment length as the benchmark. The other two longer alignments are iteratively refined by removing the furthest pair in each step (as described in Sec. 3.2) until their alignment lengths become equal to the shortest one. Then, the resulting adjusted RMSD values of the three methods are compared.

It is observed that MatAlign achieves better (smaller) adjusted RMSD values than DALI in 36 out of 67 cases (i.e. 54%), and better values than CE in 35 out of 55 cases[c] (64%). The average adjusted RMSD value for MatAlign is 1.794, and those for DALI and CE are 1.799 and 1.891, respectively. The distributions of the adjusted RMSD values for the three methods are shown in Fig. 12, and the detailed information is given in the supplementary information webpage.

4.1.4. *Speed*

In terms of speed, MatAlign is about three times faster than DALI, and about as fast as CE on Sun Ultra Sparc II with two 480 MHz CPUs and 4 GB main memory, running Sun OS 5.7. Figure 13 shows the execution times of the three methods. The average time per pairwise alignment for MatAlign is 24.8 s, and those for DALI and CE are 78.1 and 28.1 s, respectively.

[c]We are not able to extract the detailed residue–residue alignment information from 12 of the CE alignment results.

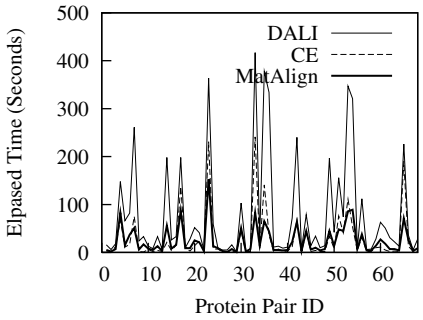


Fig. 13. Distribution of alignment times in seconds.

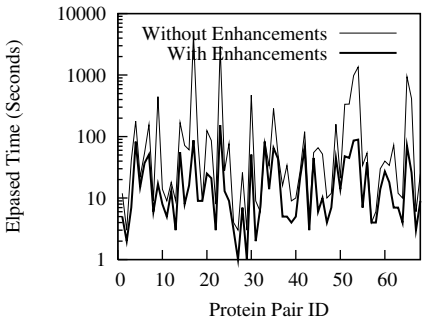


Fig. 14. Effect of speed enhancement (use of reduced rows and bands).

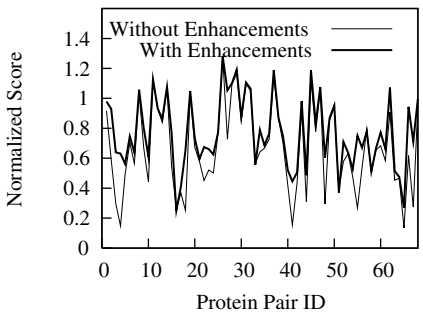


Fig. 15. Effect of accuracy enhancement (weighting of row–row matching scores and use of multiple initial alignment seeds).

4.1.5. *Significance of enhancements*

The significance of the speed and the accuracy enhancements (as described in Sec. 3.3) is shown in Figs. 14 and 15, respectively. It is observed that the speed enhancement reduces the average running time from 225.9 to 24.8 s (ninefold

speedup). Similarly, the accuracy enhancement improves the average normalized score NS from 0.67 to 0.77 (15% improvement).

4.2. *Experiment 2: on biological relevance*

In order to assess the biological meaningfulness of the results provided by MatAlign, we conduct a database search test on SCOP[20] structural domains.

We randomly select 10 proteins from Globin-like superfamily (SCOP classification `a.1.1.*`) and 10 proteins from Protein Kinases-like (PK-like) superfamily (SCOP classification `d.144.1.*`) from the representative ASTRAL data set (http://astral.berkeley.edu/) with less than 25% sequence homology. These 20 proteins are designated as the query proteins. We again randomly select 180 proteins, other than Globin-likes and PK-likes, from the same representative data set. We combine these 180 proteins with the above-mentioned 20 query proteins to form the target database of 200 proteins. (The data sets are given in the supplementary information webpage.)

We run 20 queries — taken from the Globin-like and PK-like superfamilies — against the target database. For each query, we rank the database proteins by their alignment scores (S) with respect to the query. We also conduct the same database search tests for DALI and CE. The results are ranked by their respective Z-scores and then by their RMSD values in cases of equal Z-scores.

4.2.1. *Search accuracy*

For the Globin-like queries, MatAlign achieves a perfect result. For each of the 10 queries, it can always successfully rank the Globin-like proteins in the database as the top scorers. The same result is also attained by both DALI and CE.

For the PK-like queries, MatAlign gets a nearly-perfect result. It ranks the PK-like proteins as the top scorers in 9 out of 10 query cases. Even in the incorrect case, it only ranks a single non-PK-like protein among the top 10 scorers. (For the query `1ia9A` (SCOP classification `d.144.1.5`), MatAlign incorrectly ranks a non-PK-like protein `1a48_` (SCOP classification `d.143.1.1`) at the fifth position.)

DALI as well achieves a very similar nearly-perfect result for the PK-like queries. (It also incorrectly ranks `1a48_` at the ninth position for the query `1ia9A`.) But, the accuracy of CE for the PK-like queries is inferior to those of both MatAlign and DALI. It incorrectly ranks at least one non-PK-like protein among the top 10 scorers in all but one cases. In the worst case, it ranks 7 non-PK-like proteins among the top 10 scorers. The accuracy comparison of DALI, CE and MatAlign is summarized in Table 2 and detailed in the supplementary information webpage.

4.2.2. *Speed*

The run-time statistics for DALI, CE and MatAlign for the 20 queries on the database of 200 proteins are shown in Table 3. The results are more or less similar

Table 2. Number of incorrect (non-PK-like) proteins among the top 10 scorers for the PK-like queries.

Method	1apmE	1cjaA	1csn_	1e8xA	1ia8A	1ia9A	1j7lA	1jvpP	1qpcA	1tkiA
DALI	0	0	0	0	0	1	0	0	0	0
CE	1	1	1	2	1	7	0	1	1	1
MatAlign	0	0	0	0	0	1	0	0	0	0

Table 3. Time statistics of DALI, CE and MatAlign for 20 queries on 200 proteins. (DALI's time statistics are only for the valid alignments.)

Method	Total time (hh:mm:ss)	Average time per query (hh:mm:ss.mm)	Average time per alignment (hh:mm:ss.mmmm)
DALI	37:36:04	02:34:15.66	00:00:46.2783
CE	31:26:42	01:34:20.10	00:00:28.3005
MatAlign	25:45:53	00:17:17.65	00:00:23.1883

to those of the previous experiment in Sec. 4.1. Here, MatAlign is two times faster than DALI, and has about the same speed as CE. (It should be noted that among all the $20 \times 200 = 4000$ alignments, DALI does not produce any alignment results for 1075 of them. Thus, we only take the time statistics for the remaining 2925 valid alignments into account.)

5. Conclusion

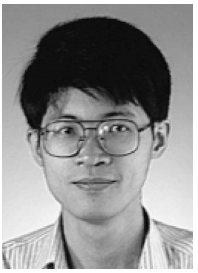
In this paper, we have presented a new scheme for comparing 3D protein structures based on the alignment of distance matrices. The experimental results show that the method is accurate and biologically relevant. It will be useful to biologists for the detailed comparative analysis of protein structures. As MatAlign has been shown to have achieved the better results than the established methods such as DALI and CE in a majority of our test cases, it can be used as an alternative tool or at least as a supplementary tool along with those established ones. As a future work, we will try to implement MatAlign as a parallel system with a view to improve its speed.

References

1. Orengo CA, Jones DT, Thornton JM, *Bioinformatics: Genes, Proteins and Computers*, BIOS Scientific Oxford, 2003.
2. Holm L, Sander C, Protein structure comparison by alignment of distance matrices, *J Mol Biol* **233**(1):123–138, 1993.
3. Shindyalov IN, Bourne PE, Protein structure alignment by incremental combinatorial extension (CE) of the optimal path, *Protein Eng* **11**(9):739–747, 1998.
4. Taylor WR, Orengo CA, Protein structure alignment, *J Mol Biol* **208**:1–22, 1989.
5. Nussinov R, Wolfson HJ, Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques, *Proc Natl Acad Sci USA* **88**:10495–10499, 1991.

6. Gibrat JF, Madej T, Bryant H, Surprising similarities in structure comparison, *Curr Opin Struct Biol* **6**:377–385, 1996.
7. Gerstein M, Levitt M, Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures, *Proc 4th Intl Conf Intell Syst Mol Biol (ISMB'96)*, pp. 59–67, 1996.
8. Akutsu T, Protein structure alignment using a graph matching technique, *Proc Genome Inform Wksp 1995 (GIW'95)*, pp. 1–8, 1995.
9. Koehl P, Protein structure similarities, *Curr Opin Struct Biol* **11**:348–353, 2001.
10. Novotny M, Madsen D, Kleywegt GJ, Evaluation of protein fold comparison servers, *Proteins Struct Funct Genet* **54**:260–270, 2004.
11. Aung Z, Tan KL, Rapid 3D protein structure database searching using information retrieval techniques, *Bioinformatics* **20**:1045–1052, 2004.
12. Kolodny R, Koehl P, Levitt M, Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures, *J Mol Biol* **346**:1173–1188, 2005.
13. Alexandrov NN, Fischer D, Analysis of topological and nontopological structural similarities in the PDB: New examples with old structures, *Proteins Struct Funct Genet* **25**:354–365, 1996.
14. Kleywegt GJ, Jones A, Superposition. *CCP4/ESF-EACBM Newslett Protein Crystall*, **31**:9–14, 1994.
15. Subbiah S, Laurents DV, Levitt M, Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core, *Curr Biol* **3**:141–148, 1993.
16. Setubal JC, Meidanis J. *Introduction to Computational Biology*, PWS Publishing, 1997.
17. Wu Z, Protein structure determination and dynamic simulation, ISU Summer Institute on Bio-informatics lecture notes, 2003. http://www.math.iastate.edu/wu/LectureNotes/SummerInstitute.ppt.
18. Holm L, Park J, DaliLite workbench for protein structure comparison, *Bioinformatics* **16**(6):566–567, 2000.
19. Fischer D, Elofsson A, Rice D, Eisenberg D, Assessing the performance of fold recognition methods by means of a comprehensive benchmark, *Proc 1996 Pacific Symp Biocomput* pp. 300–318, 1996.
20. Hubbard TJP, Ailey B, Brenner SE, Murzin AG, Chothia C, SCOP: A structural classification of proteins database, *Nucl Acids Res* **25**(1):236–239, 1997.

Zeyar Aung received his Bachelor of Computer Science (Honors) degree in 1999. He is currently a research engineer at the Institute for Infocomm Research, Singapore. He is also a Ph.D. candidate at the School of Computing of the National University of Singapore. His research interests include bioinformatics, algorithms, data mining, and database indexing and searching.



Kian-Lee Tan received his Ph.D. in computer science in 1994. He is currently an Associate Professor in the Department of Computer Science, School of Computing, National University of Singapore. His current research interests include multimedia information retrieval, query processing and optimization in multiprocessor and distributed systems, and database performance, database security and genome databases.

He was a Senior Scientist at the Genome Institute of Singapore (Joint appointment, June 01–June 03). Kian-Lee is a member of ACM and an affiliate member of IEEE.