Multiple Structure Alignment and Consensus Identification for Proteins

Jieping Ye[1], Ivaylo Ilinkin[2], Ravi Janardan[3], and Adam Isom[3]

[1] Arizona State University, Tempe, AZ 85287, USA
jieping.ye@asu.edu
www.public.asu.edu/~jye02/
[2] Rhodes College, Memphis, TN 38112, USA
ilinkin@rhodes.edu
www.rhodes.edu/MathematicsandComputerScience/FacultyandStaff/ilinkin.cfm
[3] University of Minnesota, Minneapolis, MN 55455, USA
{janardan, aisom}@cs.umn.edu
www.cs.umn.edu/~{janardan, aisom}

Abstract. An algorithm is presented to compute a multiple structure alignment for a set of proteins and to generate a consensus structure which captures common substructures present in the given proteins. The algorithm is a heuristic in that it computes an approximation to the optimal alignment that minimizes the sum of the pairwise distances between the consensus and the transformed proteins. A distinguishing feature of the algorithm is that it works directly with the coordinate representation in three dimensions with no loss of spatial information, unlike some other multiple structure alignment algorithms that operate on sets of backbone vectors translated to the origin; hence, the algorithm is able to generate true alignments. Experimental studies on several protein datasets show that the algorithm is quite competitive with a well-known algorithm called CE-MC. A web-based tool has also been developed to facilitate remote access to the algorithm over the Internet.

1 Introduction

Proteins are macromolecules that regulate all biological processes in a cellular organism [1]. The human body has about one hundred thousand different proteins that control functions as diverse as oxygen transport, blood clotting, tissue growth, immune system response, inter-cell signal transmission, and the catalysis of enzymatic reactions.

Proteins are synthesized within the cell and immediately after its creation each protein folds spontaneously into a three-dimensional (3D) configuration that is determined uniquely by its constituent amino acid sequence [14]. It is this 3D structure that ultimately determines the function of a protein, be it the catalysis of a reaction or the growth of muscle tissue or arming the body's immune system. Indeed, it is the case that where proteins are concerned "function follows form" [13].

Proteins have evolved over time through the modification and re-use of certain substructures that have proven successful [7]. It is well known [9] that during this

P. Bücher and B.M.E. Moret (Eds.): WABI 2006, LNBI 4175, pp. 115–125, 2006.
© Springer-Verlag Berlin Heidelberg 2006

process, structure is better conserved than sequence, i.e., proteins that are related through evolution tend to have similar structures even though their sequences may be quite different. Thus, the ability to identify common substructures in a set of proteins could yield valuable clues to their evolutionary history and function. This motivates the *multiple structure alignment problem* that we consider in this paper: Informally, given a collection of protein structures, we seek to align them in space, via rigid motions, such that large matching substructures are revealed. A further goal is to extract from this alignment a consensus structure that can serve as a proxy for the whole set, and could be used, for instance, as a template to perform fast searches through protein structure databases, such as the PDB, to identify structurally similar proteins.

1.1 Contributions

In this paper, we present an algorithm to compute a multiple structure alignment for a set of proteins and to generate a consensus structure. Our algorithm computes an approximation to the optimal multiple structure alignment, i.e., one that minimizes the so-called *Sum-of-Consensus distance (SC-distance)* between the consensus structure and the given set of proteins. (A more precise definition is given in Section 2.)

Our algorithm represents the input proteins and the consensus as sequences of triples of coordinates of the alpha-carbon (or C_α) atoms along the backbone. It then computes a correspondence between the coordinate triples of the C_α atoms in the different protein structures, by choosing one of the proteins as the initial consensus and applying an algorithm that is analogous to the center-star method for multiple sequence alignment [7]. Next, it derives a set of translation and rotation matrices that are optimal for the computed correspondence and uses these to align the structures in space via rigid motions and obtain the new consensus. The process is repeated until the change in SC-distance is less than a prescribed threshold. The computation of the optimal translations and rotations and the new consensus is itself an iterative process that both uses the current consensus and generates simultaneously a new one. A distinguishing feature of our algorithm is that it works directly with the coordinate representation in three dimensions with no loss of spatial information, unlike some other multiple structure alignment algorithms that operate on sets of backbone vectors translated to the origin; hence, the algorithm is able to generate true alignments.

As discussed in Section 1.2 below, there are many algorithms for multiple structure alignment. In general, it is difficult to make comparisons among them as they all operate under different sets of assumptions and problem formulations. Nonetheless, we have been able to compare our algorithm to one recent algorithm, called CE-MC [6], which also works with coordinate triples, but employs a different objective function. As discussed in Section 4, the two algorithms are very comparable in terms of the sizes of the so-called conserved regions they discover in several datasets drawn from a well-known database called HOMSTRAD [11]; however, our algorithm runs about two orders of magnitude faster than CE-MC, hence it can potentially scale to much larger datasets.

We have also incorporated our algorithm into a web-based tool to facilitate remote access and experimentation via the Internet.

Due to space constraints, we omit several details and proofs here. These can be found in the full paper [19].

1.2 Related Work

Some prior work on multiple structure alignment includes [3,4,6,10,12,20]. Orengo and Taylor [12] give algorithms for aligning pairs of proteins using a 2-level dynamic programming approach, and obtain a multiple alignment from this by aligning the pairs according to their similarity scores. Leibowitz *et al.* [10] use the technique of geometric hashing to compute a multiple alignment and core; unlike most other algorithms, theirs does not require an ordered sequence of atoms along the protein backbone. Gerstein and Levitt [4] use an iterative dynamic programming method to compute a multiple structure alignment. The CE-MC algorithm by Guda *et al.* [6], mentioned earlier, uses Monte Carlo optimization techniques to refine an initial alignment found by pairwise structure alignment using the Combinatorial Extension algorithm in [15]. Recently, Chew and Kedem [3] and Ye and Janardan [20] have shown how to compute both a multiple structure alignment and a consensus structure. In [3,20], proteins are represented as sets of unit vectors at the origin. An iterative, dynamic programming-based method is used to compute the alignment and consensus. However, a limitation of the algorithms in [3,20], is that they require the backbone vectors to be translated to the origin, hence information about the relative positions of the C_α atoms in $\Re^3$ is lost. As a result, it is not possible to generate a true alignment. By contrast, the algorithm proposed in this paper retains spatial information by representing each protein as a sequence of coordinate triples in $\Re^3$ and is able to generate a true alignment.

An important special case of multiple structure alignment is pairwise structure alignment, which involves aligning only two protein structures. Indeed, the problem arises in this paper when computing correspondences between different proteins. We use here a variant of an algorithm that we have developed recently for pairwise structure alignment [18]. Some other algorithms for pairwise alignment that may also be used (with minor modifications) include LOCK [16], DALI [8], CE [15], and the method in [2].

2 Multiple Structure Alignment: Problem Formulation

Let $\mathcal{S} = \{P_i\}_{i=1}^K$ be a set of K proteins. Protein P_i, of length L_i, consists of a chain of C_α atoms, numbered $1, 2, \cdots, L_i$, along the backbone in $\Re^3$. (As is customary [8,16], we consider only the backbone, not the amino acid residues.) We represent P_i as a sequence of *coordinate triples* $\boldsymbol{u}_j^i = (x_j^i, y_j^i, z_j^i)$, $1 \leq j \leq L_i$, on the backbone, where x_j^i, y_j^i, and z_j^i, are the coordinates of the jth C_α atom of P_i. Let $P_0 = \boldsymbol{u}_1^0, \cdots, \boldsymbol{u}_{L_0}^0$ be the *consensus structure*, of length L_0.

A *correspondence*, $\mathcal{C}$, of the K proteins $\{P_i\}_{i=1}^K$ and the consensus structure P_0 can be represented as a matrix $H = (\boldsymbol{h}_{ij})_{0 \leq i \leq K, 1 \leq j \leq L}$, for some

$L \geq \max_{0 \leq i \leq K}\{L_i\}$, where $\boldsymbol{h}_{ij}$ is either a coordinate triple belonging to the ith protein, $i \geq 0$, or a *gap*[1]. Distances between coordinate triples are based on the squared Euclidean distance between them in $\Re^3$. The distance between a coordinate triple and a gap is called a *gap penalty*, and is denoted by ρ.

A *multiple structure alignment*, $\mathcal{M}$, of the K proteins based on the correspondence $\mathcal{C}$, can be represented as another matrix $G = (\boldsymbol{g}_{ij})_{0 \leq i \leq K, 1 \leq j \leq L}$, where the ith row, for $i > 0$, is obtained via transformation (i.e., rotation and translation) of the corresponding row of H. More specifically, we combine $\{\boldsymbol{h}_{ij}\}_{j=1}^{L}$ from the ith protein, i.e., the ith row of the matrix H, into a column vector $H_i \in \Re^{L \times 3}$ as below. G_i is defined similarly from the matrix G.

$$H_i = \begin{pmatrix} \boldsymbol{h}_{i1} \\ \vdots \\ \boldsymbol{h}_{iL} \end{pmatrix} \in \Re^{L \times 3} \quad \text{and} \quad G_i = \begin{pmatrix} \boldsymbol{g}_{i1} \\ \vdots \\ \boldsymbol{g}_{iL} \end{pmatrix} \in \Re^{L \times 3}, \quad \text{for} \quad i = 1, \cdots, K.$$

Then, $G_i = (H_i - T_i) \cdot R_i = (H_i - e \cdot t_i) \cdot R_i$, for $i > 0$, where $R_i \in \Re^{3 \times 3}$ is the rotation matrix, $e \in \Re^{L \times 1}$ is a vector with 1 in each entry, $t_i \in \Re^{1 \times 3}$ is a translation vector, and $T_i = e \cdot t_i \in \Re^{L \times 3}$ is the translation matrix. The transformation of a gap is chosen to remain a gap. Note that P_0 remains unchanged; that is, $G_0 = H_0$.

Under the multiple structure alignment $\mathcal{M}$, we define the *distance between the consensus structure P_0 and protein P_j* as $D^{\mathcal{M}}(P_0, P_j) = \sum_{\ell=1}^{L} d(\boldsymbol{g}_{0\ell}, \boldsymbol{g}_{j\ell})^2$, where $d(\cdot, \cdot)$ denotes the following distance function:

$$d(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} ||\boldsymbol{u} - \boldsymbol{v}||_2, & \text{if both } \boldsymbol{u} \text{ and } \boldsymbol{v} \text{ are coordinate triples.} \\ \rho, & \text{if only one of } \boldsymbol{u} \text{ and } \boldsymbol{v} \text{ is a coordinate triple vector.} \\ 0, & \text{if both } \boldsymbol{u} \text{ and } \boldsymbol{v} \text{ are gap vectors.} \end{cases}$$

The distance between P_0 and P_j can be represented compactly as $D^{\mathcal{M}}(P_0, P_j) = ||G_0 - G_j||_F^2$, where $|| \cdot ||_F$ denotes the *Frobenius norm* [5], with the additional convention that the squared difference between a coordinate triple and a gap is ρ^2. The total distance of the K proteins to the consensus structure under $\mathcal{M}$, called the *Sum-of-Consensus distance*, or *SC-distance*, is then defined as

$$SC(\mathcal{M}) = \sum_{1 \leq j \leq K} D^{\mathcal{M}}(P_0, P_j) = \sum_{1 \leq j \leq K} ||G_0 - G_j||_F^2. \tag{1}$$

Intuitively, the *SC-distance* measures how well the consensus structure represents the given set of K proteins. A similar distance function is used in [3], where each protein is represented as a set of vectors in $\Re^4$.

Note that $G_i = (H_i - T_i) \cdot R_i$, for some rotation and translation matrices R_i and T_i. We can now define our multiple structure alignment problem as follows: *Given a set $\mathcal{S} = \{P_1, \cdots, P_K\}$ of protein structures, compute a transformation (i.e., rotation and translation) for each protein, and generate a consensus structure*

[1] In our implementation, a gap is represented by a special symbol.

P_0, such that the resulting multiple structure alignment, $\mathcal{M}$, has minimum SC-distance, $SC(\mathcal{M})$, as defined in Equation (1).

In Section 3, we present a heuristic for this problem which approximates the global minimum of the SC-distance, $SC(\mathcal{M})$, by iterative refinement of an initial multiple structure alignment and converges to a local minimum.

3 Multiple Structure Alignment: The Algorithm

Our algorithm works iteratively, by starting with an initial multiple structure alignment and updating it incrementally with decreasing SC-distance. The algorithm finally stops at some local minimum. The expectation is that with a good starting alignment, the final alignment will be close to the optimal solution. The pseudo-code for the algorithm is given as **Algorithm MAPSCI**, which stands for **M**ultiple **A**lignment of **P**rotein **S**tructures and **C**onsensus **I**dentification. The various steps are discussed in detail below.

Algorithm MAPSCI: Multiple Alignment of Protein Structures and Consensus Identification

1. Choose initial consensus structure P_0^0 from $\{P_i\}_{i=1}^K$. $i \leftarrow 0$. $SC^0 \leftarrow \infty$.
2. Do
3. if $i = 0$ then compute pairwise structure alignment between P_0^i and every P_j.
4. else use standard dynamic programming to align P_0^i with every P_j.
5. $i \leftarrow i + 1$.
6. Compute correspondence $\mathcal{C}^i$ from the above alignments (either pairwise or dynamic programming) using center-star-like method.
7. Compute optimal translation matrix T_j^i and optimal rotation matrix R_j^i iteratively, as in Section 3.4. Transform P_j by R_j^i and T_j^i for every j to obtain multiple structure alignment $\mathcal{M}^i$. $SC^i \leftarrow SC(\mathcal{M}^i)$.
8. Post-process $\mathcal{M}^i$ by removing all columns consisting of only gaps.
9. Compute new consensus structure P_0^i from $\mathcal{M}^i$ by Theorem 1.
10. Until $\left|\frac{SC^i - SC^{i-1}}{SC^{i-1}}\right| \leq \eta$. // η is a user-specified threshold

3.1 Step I (Line 1): Choose the Initial Consensus Structure

There are many ways to choose the initial consensus structure P_0^0. One possibility is to choose P_0^0 as the *center protein*, so that it minimizes the sum of the minimum pairwise distances to all the other proteins. That is, P_0^0 is the $k^\star$th protein, where $k^\star = \arg\min_{1 \leq k \leq K} \sum_{i=1}^K D(P_k, P_i)$, and $D(P_k, P_i)$ denotes the distance between P_k and P_i, as computed by a pairwise structure alignment.

This choice makes sense intuitively, since it yields a consensus structure which is "not too far away" from the others; however it is expensive computationally, as it involves $\frac{K(K-1)}{2}$ pairwise alignments. A simple and less expensive choice that appears to work well is to pick P_0^0 as the *median protein*, i.e., the protein of median length. We report our experimental results for both choices in Section 4.

3.2 Step II (Lines 3 and 4): Pairwise Structure Alignment

After we determine the consensus structure P_0^0 in Step I, the $K-1$ pairwise structure alignments between P_0^0 and $P_i \neq P_0^0$, for $i = 1, \cdots, K$, are computed using the pairwise structure alignment algorithm in [18]. (Other pairwise structure alignment algorithms, such as LOCK [16], DALI [8], CE [15], etc. could also be used instead.) After the initial step, the consensus structure is expected to be close to the proteins in the dataset. Dynamic programming on the coordinates of the C_α atoms is then applied to compute the alignment (see Line 4).

3.3 Step III (Line 6): Compute an Initial Correspondence

An initial correspondence between the K proteins is obtained by applying to the consensus structure and the pairwise alignments computed in Steps I and II a method adapted from the center-star method [7] for multiple sequence alignment. We call our extension of this method to protein structures a *center-star-like* method. There are two key differences between the two methods. First, in the center-star-like method we will be aligning not alphabet characters representing amino acids but coordinate triples derived from the protein backbones. Second, in a multiple structure alignment, the correspondence computed using the center-star-like method is only a first step; it is followed by an optimization step to compute the optimal transformation matrices. Subsequent correspondences are computed similarly.

3.4 Step IV (Lines 7 and 9): Compute Optimal Rotation and Translation Matrices and Consensus Structure

Given a correspondence $\mathcal{C}$ and a consensus structure J, we show how to find both the optimal rotation matrix R_j and translation matrix T_j for each protein P_j as well as the new consensus structure $\bar{J}$.

Assume the correspondence $\mathcal{C}$ is represented as a matrix $H = (\boldsymbol{h}_{ij})$. Protein P_j in $\mathcal{C}$ can be represented as H_j consisting of $\{\boldsymbol{h}_{ji}\}_{i=1}^{L}$, where $\boldsymbol{h}_{ji}$ represents either a coordinate triple or a gap. The objective is to find the rotation and translation matrices R_j and T_j, for $j = 1, \cdots, K$, and the consensus structure $\bar{J}$, such that sum of the pairwise alignment distances between $\bar{J}$ and each (transformed) P_j is minimum.

Mathematically, given H_j, for $j = 1, \cdots, K$, we wish to compute optimal rotation and translation matrices R_j and T_j, for each protein P_j, and the consensus structure $\bar{J}$, such that $S = \sum_{j=1}^{K} ||\bar{J} - (H_j - T_j) \cdot R_j||_F^2$ is minimized.

Direct minimization of S over $\bar{J}$, and the T_j's and R_j's seems difficult. Instead, we propose an iterative procedure for minimizing S. Within each iteration, the minimization of S is carried out in two stages that are interleaved: computation of the optimal $\bar{J}$ for given R_j's and T_j's; and computation of the optimal R_j's and T_j's for a given $\bar{J}$.

First, we show how to compute the consensus structure, given the rotation and translation matrices R_j's and T_j's, as stated in the following theorem:

Theorem 1. *Assume that the correspondence $\mathcal{C}$ is represented as a matrix $H = (\boldsymbol{h}_{ij})$ and $\bar{J} = (J_1, \cdots, J_L)^T$ is the optimal consensus structure. For each column j, let I_n be the set of indices of proteins with a non-gap in the jth column and I_g be the set of indices of proteins with a gap in the jth column. Then $\bar{J}_j$, the jth position of the optimal consensus structure, is either a coordinate triple $\boldsymbol{u}_j$, where $\boldsymbol{u}_j = \frac{1}{|I_n|} \sum_{i \in I_n} \boldsymbol{h}_{ij}$, or a gap.*

Next, we show how to compute the optimal translation matrix T_i, for each i, for a given consensus structure $\bar{J}$. From the equation above for S, it is clear that the optimal T_i and T_j, for $i \neq j$ are independent of each other. Hence, we focus on the computation of T_i, for a specific i. The translation matrix T_i can be decomposed as $T_i = e \times t_i$, where $t_i \in \Re^{1\times 3}$ is the translation vector.

As mentioned earlier, the transformation of a gap remains a gap. Hence the computation of the translation and rotation matrices is independent of the mismatches (i.e., where at least one of the two elements being compared is a gap). We can thus simplify the computation by removing all mismatches in the alignment between the consensus structure $\bar{J}$ and the ith protein P_i. Let $A \in \Re^{n\times 3}$ and $B \in \Re^{n\times 3}$ consist of the coordinate triples from the consensus structure and the ith protein, respectively, after removing the mismatches. Here n is the number of matches between the consensus structure and the ith protein (i.e., comparison of two non-gaps).

The optimal rotation matrix can be obtained by computing the Singular Value Decomposition (SVD) [5] of $A^T B$ as in [17], while it is known that the optimal translation vector is the one that matches the centroids of the coordinate triple vectors from A and B as follows:

Theorem 2. *Let A and B be defined as above. Assume that $e^T A = [0, 0, 0]$. Then for any rotation matrix R_i, the optimal translation vector t_i for minimizing $S_i = ||A - (B - T_i) \cdot R_i||_F^2 = ||A - (B - e \cdot t_i) \cdot R_i||_F^2$ is given by $t_i = \frac{1}{n} e^T B$.*

3.5 Analysis

We show in the full paper [19] that, in **Algorithm MAPSCI**, the SC-distance is non-increasing from one iteration to the next and the algorithm thus converges.

Let ℓ be the maximum length of the K proteins, m_1 the number of iterations in computing the optimal rotation and translation matrices, and m_2 the number of iterations of the loop in lines 2–10. The time complexity of the algorithm can be shown to be either $O(K^2\ell^2 + (K\ell^2 + K^2\ell m_1)m_2)$ or $O(K + (K\ell^2 + K^2\ell m_1)m_2)$, depending on the choice of the initial consensus. In practice, m_1 and m_2 tend to be small constants, so the running time is either $O(K^2\ell^2)$ or $O(K\ell^2 + K^2\ell)$.

4 Experimental Results

4.1 Software and Web Server

Algorithm MAPSCI has been implemented in C, and has been tested on several protein structure datasets, as described below. Moreover, it has been

incorporated into a web-based tool (also called **MAPSCI**) for remote access over the Internet. This tool allows proteins to be specified by their PDB ids (or uploaded from local files). The results of the alignment are annotated in the JOY output format and the standard NBRF/PIR format, and can be visualized using the molecular viewer applet Chemis 3D integrated with the web tool. The software and the web-based tool can be accessed at `www.geom-comp.umn.edu` .

4.2 Datasets

We evaluated our algorithm using the seventeen datasets shown in Table 1. The ten proteins in Set 1 are from the *Globin* family, while the ten proteins in Set 2 are from the *Thioredoxin* family. Set 3 contains four all-alpha proteins, which are structural neighbors of the protein `1mbc` (*Myoglobin*) from the DALI database. The four alpha-beta proteins in Set 4 are all structural neighbors of the protein `3trx` (*Thioredoxin-Reduced Form*) from the DALI database. Sets 5–7 are from [3]: Set 5 contains sixteen proteins from the *Globin* family, Set 6 contains six all-beta proteins from the *immunoglobulin* family, and Set 7 contains five proteins that are unrelated. Sets 8–17 are from the HOMSTRAD database [11]. HOMSTRAD (HOMologous STRucture Alignment Database) is a curated database of structure-based alignments for homologous protein families. Datasets 8–17 have sequence identities ranging from 17% to 47%.

Table 1. The test datasets used for the experiments

	Dataset	Proteins (PDB ID)
1	*Globin*	1mbc, 1mba, 1dm1, 1hlm, 2lhb, 2fal, 1hbg, 1flp, 1eca, 1ash
2	*Thioredoxin*	3trx, 1aiu, 1erv, 1f9mA, 1ep7A, 1tof, 2tir, 1thx, 1quw, 1fo5A
3	*all-alpha*	1le2, 2fha, 1nfn, 1grj
4	*Alpha-beta*	1mek, 1a8l, 1f37B, 1ghhA
5	*Globin*	1hlb, 1hlm, 1babA, 1babB, 1ithA, 1mba, 2hbg, 2lhb 3sdhA, 1ash, 1flp, 1myt, 1eca, 1lh2, 2vhbA, 5mbn
6	*all-beta*	1cd8, 1ci5A, 1qa9A, 1cdb, 1neu, 1qfoA
7	*mixed*	1cnpB, 1jhgA, 1hnf, 1aa9, 1eca
8	*biotin_lipoyl*	1bdo, 1lac, 1ghk,1iyv, 1pmr, 1qjoA, 1fyc
9	*msb*	1esl, 1rdol, 1msbA, 1tn3, 2afp, 1qddA, 1ixxA, 1ixxB
10	*cyt3*	2cdv, 2cym, 1wad, 3cyr, 2cy3, 1aqe
11	*ghf7*	1celA, 1eg1A, 2ovwA, 2a39A
12	*Acetyltransf*	1bo4A, 1cm0A, 1yghA, 1qstA, 1b87A, 1cjwA
13	*HMG_bo*	1hrzA, 1nhn, 1cktA, 1hma, 2lefA
14	*intb*	1afcA, 2afgA, 2fgf, 2mib, 1i1b, 1iraX
15	*lipocalin*	1i4uA, 1bbpA, 1exsA, 1bebA, 2a2uA, 1jv4A, 1ew3A, 1bj7, 1e5pA, 1hn2, 1dzkA, 1iiuA, 1aqb, 1epaA, 1qqsA
16	*sh3*	1awj, 1shg, 1shfA, 2src, 1qcfA, 1lckA, 1qlyA, 1bbzA, 1griA, 1gbrA, 1cskA, 1ckaA, 1semA, 1griA, 1ycsB, 2hsp, 1ark, 1aojA, 1pht, 1bb9
17	*TPR*	1a17, 1elwA, 1elrA, 1e96B, 1fchA, 1ihgA

4.3 Results

Our first experiment compared two choices for the initial consensus: "center" and "median". The results for the seventeen datasets are illustrated in Figure 1, where the x-axis denotes the different datasets and the y-axis denotes the SC-distance (left) and the running time (right). Throughout our experiments, we

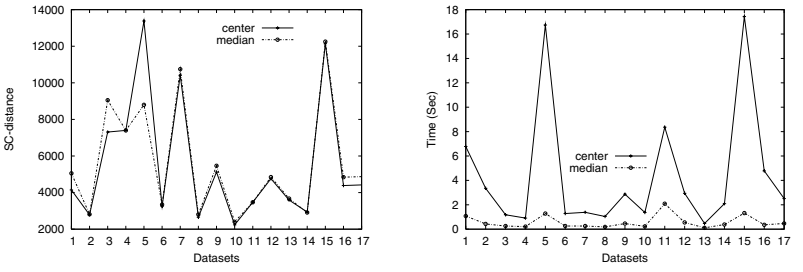


Fig. 1. Comparison of choices of initial consensus structure: "center" and "median," using the seventeen datasets. Time measurements made on a 500MHz Sun-Sparc with 512MB memory.

used $\eta = 0.1$ and $\rho = 16$. (Recall that η is a user-specified threshold for convergence—see line 10 of **Algorithm MAPSCI**; ρ is the gap penalty.)

As seen from the figure, the "center" method produced SC-distances that were slightly lower than those produced by "median" (except for Set 5). However, the computational cost of "center" is much higher than that of "median".

Our second experiment was to visualize the multiple structure alignments. An example for Set 2 is shown in Figure 2, for the two choices of the initial consensus, together with the computed final consensus. In each case, the computed consensus is seen to be visually quite similar to the input proteins. (Color versions of all the figures may be accessed at `www.geom-comp.umn.edu` .)

Our third experiment compared our method with CE-MC [6], using the manually-aligned HOMSTRAD database as the "gold" standard. Only conserved

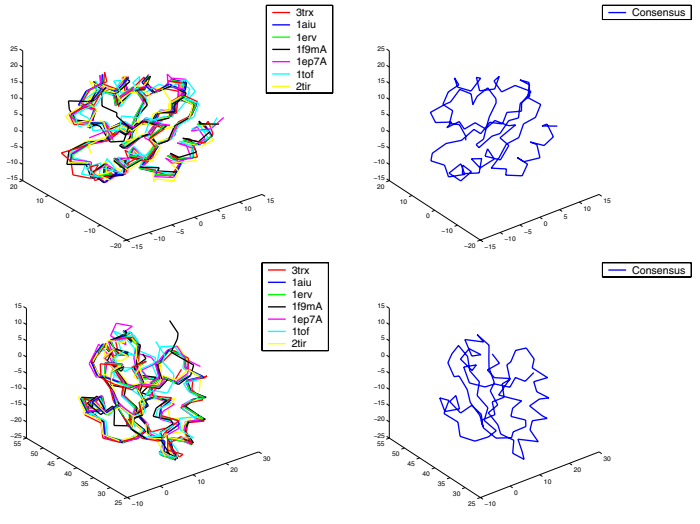
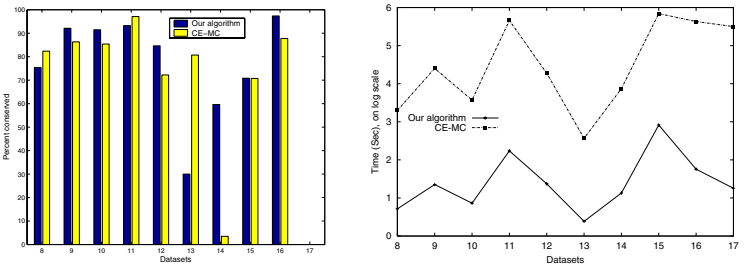


Fig. 2. Multiple structure alignment using dataset 2, with two choices for the initial consensus: "center" (top; SC-distance: 2783.47), and "median" (bottom; SC-distance: 2814.20). To avoid clutter, only the first seven proteins from the dataset are displayed.

columns (i.e., ones with no gaps) were considered for the comparison, as they are viewed to be more important biologically. We ran our algorithm and CE-MC on the ten datasets, Sets 8–17, from HOMSTRAD and reported the percentage of conserved columns from HOMSTRAD that also appeared in the alignments from these two algorithms. See Figure 3(a). Overall, our algorithm is seen to be quite competitive with CE-MC, in terms of the sizes of the conserved regions. Figure 3(b) shows the running times for both algorithms on a log scale, i.e., $\ln(1+t)$ is plotted for each t. The running time of our algorithm is seen to be much smaller (about two orders of magnitude) than that of CE-MC, implying that it can potentially scale to much larger datasets than can CE-MC.



(*a*) (*b*)

Fig. 3. Comparison of our algorithm and CE-MC on the ten datasets from HOMSTRAD: the percentage conserved (graph (a)), and the computational time (graph (b)), on a log scale. Time measurements made on a 500MHz Sun-Sparc with 512MB memory.

References

1. C. Branden, and J. Tooze. Introduction to Protein Structure, Garland, 1999.
2. L.P. Chew, K. Kedem, D.P. Huttenlocher, and J. Kleinberg. Fast detection of geometric substructure in proteins. *J. Comput. Bio.*, 6:(3-4), 1999, 313–325.
3. L.P. Chew and K. Kedem. Finding the consensus shape of a protein family. Proc. *ACM Symp. Comput. Geometry SoCG'02*, 64–73, 2002.
4. M. Gerstein and M. Levitt. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. Proc. *ISMB'96*, 59–66, 1996.
5. G.H. Golub and C.F. Van Loan. Matrix Computations, John Hopkins University Press, 3rd edition, 1996.
6. C. Guda, E.D. Scheeff, P.E. Bourne, I.N. Shindyalov. A new algorithm for the alignment of multiple protein structures using Monte Carlo optimization. Proc. *PSB'01*, 275–286, 2001.
7. D. Gusfield. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge University Press, 1997.
8. L. Holm and C. Sander. Protein Structure Comparison by Alignment of Distance Matrices. *J. Mol. Bio.*, 233, 1993, 123–138.
9. L. Holm and C. Sander. Mapping the protein universe. *Science*, 273, 1996, 595–602.

10. N. Leibowitz, Z. Fligelman, R. Nussinov, and H. Wolfson: Multiple Structural Alignment and Core Detection by Geometric Hashing. Proc. *ISMB'99*, 169–177, 1999.
11. K. Mizuguchi, C.M. Deane, T.L. Blundell, J.P. Overington. HOMSTRAD: a database of protein structure alignments for homologous families. *Prot. Sci.*, 7, 1998, 2469–2471.
12. C. Orengo and W. Taylor. SSAP: Sequential structure alignment program for protein structure comparison. *Meth. Enzymol.*, 266, 1996, 617–635.
13. G. Rose. No assembly required. *The Sciences*, 36, 1996, 26–31.
14. M. Sela, F. H. White Jr, and C. B. Anfinsen. Reductive cleavage of disulfide bridges in Ribonuclease. *Science*, 125, 1957, 691-692.
15. I.N. Shindyalov and P.E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Prot. Eng.*, 11, 1998, 739–747.
16. A.P. Singh and D.L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representation. Proc. *ISMB'97*, 284–293, 1997.
17. S. Umeyama. Least-square estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13 (4), 1991, 376–380.
18. J. Ye, R. Janardan, and S. Liu. Pairwise protein structure alignment based on an orientation-independent backbone representation. *J. Bio. Comput. Bio.*, 2 (4), 2004, 699–717.
19. J. Ye, I. Ilinkin, R. Janardan, and A. Isom. Multiple structure alignment and consensus identification for proteins. Submitted. Available at `www.geom-comp.umn.edu`.
20. J. Ye, R. Janardan. Approximate multiple protein structure alignment using the Sum-of-Pairs distance. *J. Comput. Bio.*, 11 (5), 2004, 986–1000.