Ilinkin *et al. BMC Bioinformatics* 2010, **11**:71
http://www.biomedcentral.com/1471-2105/11/71

BMC Bioinformatics

SOFTWARE Open Access

Multiple structure alignment and consensus identification for proteins

Ivaylo Ilinkin[1*], Jieping Ye[2], Ravi Janardan[3]

Abstract

Background: An algorithm is presented to compute a multiple structure alignment for a set of proteins and to generate a consensus (pseudo) protein which captures common substructures present in the given proteins. The algorithm represents each protein as a sequence of triples of coordinates of the alpha-carbon atoms along the backbone. It then computes iteratively a sequence of transformation matrices (i.e., translations and rotations) to align the proteins in space and generate the consensus. 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.

Results: Experimental results show that the algorithm converges quite rapidly and generates consensus structures that are visually similar to the input proteins. A comparison with other coordinate-based alignment algorithms (MAMMOTH and MATT) shows that the proposed algorithm is competitive in terms of speed and the sizes of the conserved regions discovered in an extensive benchmark dataset derived from the HOMSTRAD and SABmark databases.
The algorithm has been implemented in C++ and can be downloaded from the project's web page. Alternatively, the algorithm can be used via a web server which makes it possible to align protein structures by uploading files from local disk or by downloading protein data from the RCSB Protein Data Bank.

Conclusions: An algorithm is presented to compute a multiple structure alignment for a set of proteins, together with their consensus structure. Experimental results show its effectiveness in terms of the quality of the alignment and computational cost.

Background

This paper presents an algorithm to compute a multiple structure alignment for a set of proteins and to generate a consensus structure. The algorithm is called MAPSCI, which stands for **M**ultiple **A**lignment of **P**rotein **S**tructures and **C**onsensus **I**dentification. MAPSCI addresses the problem of global structure alignment, which has also been considered by CE-MC [1], MAMMOTH [2], and MATT [3]. Specifically, MAPSCI computes an approximation to the multiple structure alignment that minimizes the so-called *Sum-of-Consensus distance (SC-distance)*, i.e. the sum of the pairwise distances between the consensus structure and each protein in the set (see the **Methods** section for the precise definition of *SC-distance*). Our experiments show that MAPSCI converges quite rapidly and produces alignments that compare favorably with the alignments produced by MAMMOTH and MATT. The consensus structures generated by MAPSCI are visually quite similar to the input proteins. Although the consensus structures are not real proteins, they could be used, for instance, as templates to perform fast searches through protein structure databases, such as the Protein Data Dank [4], to identify structurally similar proteins.

MAPSCI has similar structure to the algorithm of Ye and Janardan [5]. However, MAPSCI works directly on the coordinates of the C_α atoms and produces true alignments; by contrast, the algorithm in [5] requires that the backbone vectors be translated to the origin, hence information about the relative positions of the C_α atoms in $\mathbb{R}^3$ is lost and as a result the algorithm does not generate true alignments. The **Methods** section presents the mathematical and algorithmic framework of MAPSCI and provides the complete details where the

* Correspondence: iilinkin@gettysburg.edu
[1]Department of Computer Science, Gettysburg College, Gettysburg, PA, USA

two algorithms differ significantly; when there is an overlap the reader is referred to publication [5].

Implementation

MAPSCI 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 [6]. Next, MAPSCI 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. This iterative process is well-defined as it is shown in the **Methods** section that the *SC-distance* is non-increasing from one iteration to the next. 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.

Table 1 summarizes the algorithm in pseudocode form. The various steps in the pseudocode are described in more detail in the **Methods** section. The algorithm has been implemented in C++ and can be used standalone or run remotely via a web-based interface. The source code of the implementation is available for download from the project's website (see the **Availability** section). The implementation is organized as a library of algorithms and simple data structures that can be integrated in other projects. Examples of using the library within a C++ program are given in the README file of the source code distribution. The iterative process described above employs pairwise structure alignment as an intermediate step and the parameters that control the execution of the multiple alignment algorithm are the parameters for the underlying pairwise alignment algorithm. The current implementation uses the pairwise alignment algorithm described in [7]; however, other algorithms for pairwise structure alignment can be used instead.

Results

Web Server

MAPSCI has been incorporated into a web server for remote access over the Internet (see Figure 1). This tool allows for protein structures to be uploaded from files on the local disk or retrieved from the Protein Data Bank (PDB) [4] by specifying their PDB ids. The results from the alignment are annotated in the standard NBRF/PIR format, which can be previewed online via the Jalview applet [8]. Integrated with the server is the molecular viewer applet Chemis 3D [9], which allows for visualization of the aligned protein structures.

The web server offers a simple interface that allows for remote access from within other software. Table 2 gives an example of using the programming language Python to retrieve the transformed coordinates (in PDB format) for the multiple alignment of the structures from the HOMSTRAD CUB family. Additional examples and the complete set of options for remote access can be found at the server web page (see the **Availability** section).

Comparison

As discussed earlier, there are many algorithms for multiple structure alignment. In general, it is difficult to make comparisons among them, since they operate under different sets of assumptions and problem formulations. We compare MAPSCI to two recent algorithms – MAMMOTH [2] and MATT [3] – which also work with coordinate triples, but employ a different objective function. Our experiments show that MAPSCI is competitive in terms of the sizes of the so-called conserved regions and runs significantly faster than the other two algorithms, hence can potentially scale to much larger datasets.

The comparison is based on two benchmark datasets. The first dataset is compiled from the HOMSTRAD database [10], which is a curated database of structure-based alignments for homologous protein families and is considered the "gold" standard. The benchmark dataset consists of the 232 HOMSTRAD families that have at least 4 structures. The second dataset consists of the *superfamily set* in the SABmark database [11] (version 1.65). It contains 425 families with low to intermediate sequence similarity. The metrics considered in the comparison are the *strict core* (or just *core*) and the core RMSD. This follows the experimental setup in [2] where *strict core* is defined as "the set of positions with 100% conservation, and within 4.0 Å of each other in the final structural alignment in 3D". A similar metric is discussed in [12] and [13]. The results are summarized in Figures 2 and 3, which show the pairwise comparisons (MAPSCI, MAMMOTH), (MAPSCI, MATT) in terms of the core size (expressed in percent of the length of the shortest protein) and the core RMSD. Table 3 provides a comparison of the average core size and average core RMSD for the three methods on the benchmark datasets.

In general, it is difficult to compare two algorithms based on these two metrics (larger cores tend to have larger RMSD). However, on the HOMSTRAD dataset MAPSCI outperformed MAMMOTH in 45% of the test cases and MATT in 59% of the test cases by computing

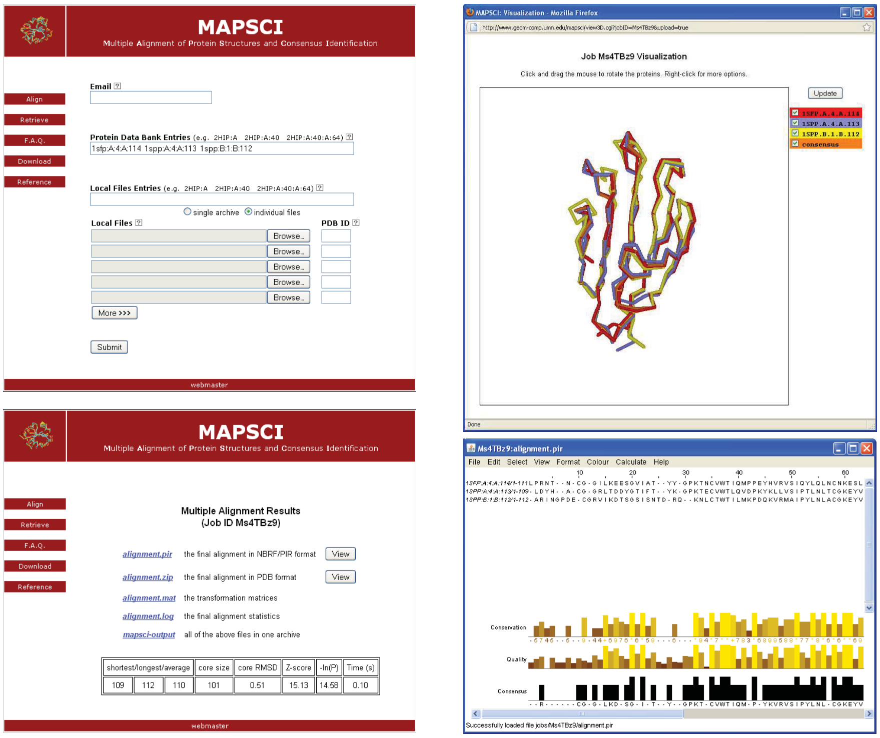
Table 1 Algorithm MAPSCI: Multiple Alignment of Protein Structures and Consensus Identification

```
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 (Theorems 2 and 3). 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 ⊠$^i$ 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 $\left|\frac{SC^i - SC^{i-1}}{SC^{i-1}}\right| \le \eta$ .//$\eta$ is a user-specified threshold (currently set at 0.0001)
```

alignments with both larger cores and smaller core RMSD. (MAMMOTH and MATT were better than MAPSCI on both metrics combined in 6% and 5% of the test cases, respectively). MAPSCI computed cores for all 232 test cases, while MAMMOTH failed to compute a core for one family (*bowman*), and MATT failed to compute a core for three families (*asp*, *lipocalin*, and *tln*).

On the SABmark dataset MAPSCI computed larger cores with better RMSD in 39% of the test cases when compared with MAMMOTH and in 37% of the test cases against against MATT. (MAMMOTH and MATT were better than MAPSCI on the two metrics combined in 15% and 26% of the test cases, respectively.) MATT was the most robust of the three algorithms and failed to compute a core in only five test cases; MAPSCI failed on 40 families and MAMMOTH failed on 31 families.

MAPSCI took only 151 seconds to align the 425 families in the SABmark dataset and 85 seconds to align the families in the HOMSTRAD dataset. MAMMOTH took 1100 seconds on the SABmark dataset and 649 seconds on the HOMSTRAD dataset. By contrast,



Email
Protein Data Bank Entries (e.g. 2HIP:A 2HIP:A:40 2HIP:A:40:A:64)
1sfp:A:4:A:114 1spp:A:4:A:113 1spp:B:1:B:112
Local Files Entries (e.g. 2HIP:A 2HIP:A:40 2HIP:A:40:A:64)
○ single archive ● individual files
Local Files
PDB ID

Multiple Alignment Results
(Job ID Ms4TBz9)

alignment.pir	the final alignment in NBRF/PIR format
alignment.zip	the final alignment in PDB format
alignment.mat	the transformation matrices
alignment.log	the final alignment statistics
mapsci-output	all of the above files in one archive

shortest/longest/average			core size	core RMSD	Z-score	-ln(P)	Time (s)
109	112	110	101	0.51	15.13	14.58	0.10

http://www.geom-comp.umn.edu/mapsci/view3D.cgi?jobID=Ms4TBz9&upload=true

Job Ms4TBz9 Visualization

Click and drag the mouse to rotate the proteins. Right-click for more options.

☑ 1SFP.A.4.A.114
☑ 1SPP.A.4.A.113
☑ 1SPP.B.1.B.112
☑ consensus

Ms4TBz9:alignment.pir

```
                               10        20        30        40        50        60
1SFP:A:4:A:114/1-111 LPRNT--N-CG-GILKEESGVIAT--YY-GPKTNCVWTIQMPPEYHVRVSIQYLQLNCNKESL
1SPP:A:4:A:113/1-109 -LDYH--A-CG-GRLTDDYGTIFT--YK-GPKTECVWTLQVDPKYKLLVSIPTLNLTCGKEYV
1SPP:B:1:B:112/1-112 -ARINGPDE-CGRVIKDTSGSISNTD-RQ--KNLCTWTILMKPDQKVRMAIPYLNLACGKEYV

                     -5745--5--9-44+6976*6*59---6---*94*7**+783*6899588*77*8*6*6**69

                     --R------CG-G-LKD-SG-I-T--Y--GPKT-CVWTIQM-P-YKVRVSIPYLNL-CGKEYV
```

Successfully loaded file jobs/Ms4TBz9/alignment.pir

Figure 1 Web server screenshots. Screenshots from the web server: main page (top left), results page (bottom left), structure view (top right), sequence view (bottom right).

Table 2 Remote access to the server

```
import urllib2
url = “http://www.geom-comp.umn.edu/mapsci/align.cgi?wsget=pdb&rcsb=1sfp+1spp:A+1spp:B”
server = urllib2.urlopen(url)
output = file("alignment.zip”, ‘wb’)
output.write(server.read())
output.close()
server.close()
```

An example of using the programming language Python to retrieve the transformed coordinates (in PDB format) for the multiple alignment of the structures from the HOMSTRAD CUB family. Additional examples and the complete set of options for remote access can be found at the server web page (see the Availability section).

MATT took several hours to process the two datasets. Figure 4 shows the actual time taken by MAPSCI for all families in the benchmark dataset in terms of the total number of residues per family. The algorithm converges very quickly and can potentially scale to large datasets. The machine used for all experiments reported in the paper runs Ubuntu Linux 8.04 and has 4 GB of RAM with Intel®Core™2 Quad CPU Q9550 @ 2.83 GHz. MAMMOTH and MATT were run with their default parameter settings.

Methods

In this section, we provide the mathematical and algorithmic framework underlying MAPSCI. As mentioned earlier MAPSCI shares common elements with the algorithm in [5], and therefore, we follow the same general outline. However, we only present the full details when there are significant differences and refer the reader to [5] when there is an overlap.

Multiple Structure Alignment: Problem Formulation

Let $\{P_1, P_2, ..., P_k\}$ be the given set of K proteins and let l_i be the number of C_α atoms along the backbone of protein P_i. We represent P_i as a sequence of *coordinate triples* $\vec{u}^{\,i}_j = (x^i_j, y^i_j, z^i_j)$, $1 \le j \le l_i$, that represent the coordinates of the jth C_α atom of P_i along the backbone. (As is customary [14,15], we consider only the backbone, not the amino acid residues themselves.) Let $P_0 = \vec{u}^{\,0}_1, ..., \vec{u}^{\,0}_{l_0}$ denote the *consensus structure*, of length l_0.

A *correspondence* of the K proteins in $\mathcal{S}$ and the consensus structure P_0 can be represented as a matrix $H = (\vec{h}_{ij})_{0\ \le\ i\ \le\ K, 1\ \le\ j\ \le\ L}$, for some $L \ge \max_{0\ \le\ i\ \le\ K}\{l_i\}$, where $\vec{h}_{ij}$ is either a coordinate triple belonging to the ith protein or a *gap*. Distances between coordinate triples are based on the squared distance between them in $\mathbb{R}^3$. The distance between a coordinate triple and a gap is called a *gap penalty*, and is denoted by ρ.

The results reported in this paper use 16.0 for the value of the gap penalty.

Let $G_i = (H_i - T_i)R_i = (H_i - e \times t_i)R_i$, for $i > 0$, where $R_i \in \mathbb{R}^{3\ \times\ 3}$ is some rotation matrix, $T_i = e \times \vec{t}_i$ is the translation matrix, $e \in \mathbb{R}^{L\ \times\ 1}$ is a vector with 1 in each entry, and $\vec{t}_i \in \mathbb{R}^{1\ \times\ 3}$ is a translation vector. (The transformation of a gap remains a gap.) Note that P_0 remains unchanged, i.e. $G_0 = H_0$.

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

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

The distance between P_0 and P_j can be represented compactly as $D(P_0, P_j) = ||\, G_0 - G_j \,||^2_F$, where $||\cdot||_F$ denotes the *Frobenius norm* [16], 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, called the *Sum-of-Consensus distance*, or *SC-distance*, is then defined as

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

Table 3 Benchmark datasets performance

	HOMSTRAD		SABmark	
	Average Core (%)	Average Core RMSD	Average Core (%)	Average Core RMSD
MAPSCI	70.99	$0.83_{(n\ =\ 232)}$	48.89	$1.00_{(n\ =\ 385)}$
MAMMOTH	66.74	$0.83_{(n\ =\ 231)}$	44.55	$0.99_{(n\ =\ 394)}$
MATT	63.79	$0.85_{(n\ =\ 229)}$	47.88	$0.99_{(n\ =\ 420)}$

Statistics for the performance of the three methods on the benchmark datasets. The subscripts in the *Average Core RMSD* columns indicate how many values were used in computing the statistics, since the algorithms failed to compute a core for some of the data sets. For the *Average Core (%)* columns all reported values were used and therefore $n = 232$ and $n = 425$ for the HOMSTRAD and SABmark datasets, respectively.

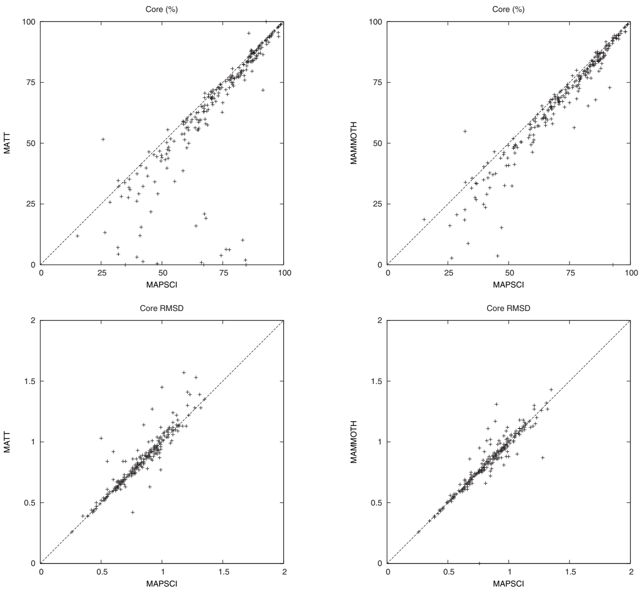


Figure 2 HOMSTRAD dataset comparison. Comparison based on the *strict core* metric (expressed in percent of the size of the shortest protein) and the *strict core RMSD* on the HOMSTRAD dataset.

Intuitively, the *SC-distance* measures how well the consensus structure represents the given set of K proteins. A similar distance function is used in [17], where each protein is represented as a set of vectors in $\mathbb{R}^4$.

We can now define the multiple structure alignment problem as follows:

Multiple Structure Alignment Problem

Given a set $\{P_1, P_2, ..., P_K\}$ of protein structures, compute a transformation (i.e., rotation and translation) for each protein, and generate a consensus structure P_0, such that the resulting multiple structure alignment has minimum SC-distance as defined in Equation (1).

In the next section, we present a heuristic for this problem. Our algorithm approximates the global minimum of the *SC-distance* by iterative refinement of an initial multiple structure alignment and converges to a local minimum.

Step I: Choice of the initial consensus structure

We consider four choices for initial consensus structure: (i) *median protein*, i.e. the protein of median length; (ii) *center protein*, i.e. the protein that minimizes the sum of the pairwise distances to all the other proteins; (iii) the *minmax* protein, i.e. the protein with the smallest maximum pairwise distance; and (iv) *maxcore protein*, i.e. the protein that generates the largest initial core. (The first three choices for initial consensus are considered in [5].)

The experimental results in Figure 5 indicate that MAPSCI is quite robust in terms of the choice of initial consensus. However, the data suggests that the *median protein* occasionally leads to alignments with very low core size, and therefore is the least reliable choice. The other three choices seem to work well in practice, although they are more expensive computationally. The results reported in the **Comparison** section use the *maxcore protein* as the initial consensus.

Step II: Compute an initial correspondence

After we determine the consensus structure P_0 in Step I, the K - 1 pairwise structure alignments between P_0 and $P_i \neq P_0$, for i = 1, ..., K, are computed using the algorithm in [7]. (Other pairwise structure alignment algorithms could also be used instead.) The K - 1 pairwise structure are combined in Line 6 of the algorithm (Table 1) using the *center-star-like* method described in [5].

Step III: Compute optimal rotation and translation matrices and consensus structure

Given a correspondence $H = (\vec{h}_{ij})$ the objective is to find the rotation and translation matrices R_j and T_j, for j = 1, ..., K, and the consensus structure $\overline{J}$, such that the sum of the pairwise alignment distances between $\overline{J}$ and each (transformed) P_j is minimum; i.e. we wish to minimize

$$S = \sum_{1\leq j\leq K} || \overline{J} - (H_j - T_j)\cdot R_j ||_F^2. \qquad (2)$$

Direct minimization of S over $\overline{J}$, 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: (1) computation of the optimal $\overline{J}$ for given

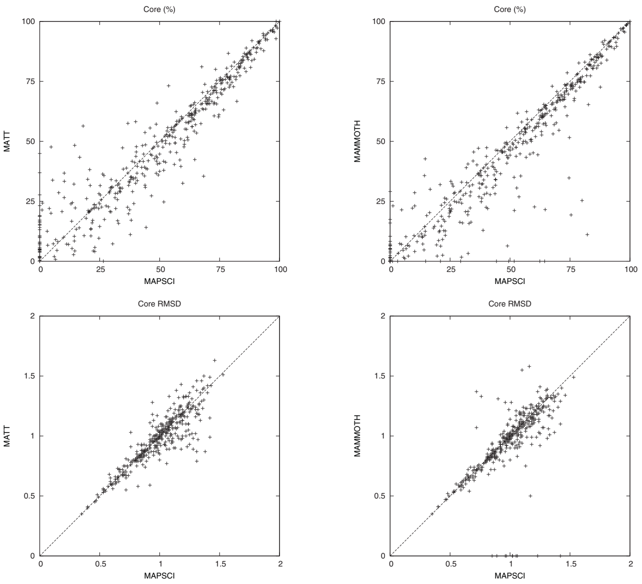


Figure 3 SABmark dataset comparison. Comparison based on the *strict core* metric (expressed in percent of the size of the shortest protein) and the *strict core RMSD* on the SABmark dataset.

R_j's and T_j's, and (2) computation of the optimal R_j's and T_j's for a given $\overline{J}$.

Computation of the optimal consensus structure

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 is represented as a matrix* $H = (\vec{h}_{ij})$ *and* $\overline{J} = (J_1, ..., J_L)^T$ *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* $\overline{J}_j$*, in the jth position of the optimal consensus structure equals either the coordinate triple* $x_j = \frac{1}{|I_n|}\sum_{i\in I_n}\vec{h}_{ij}$ *, or a gap.*

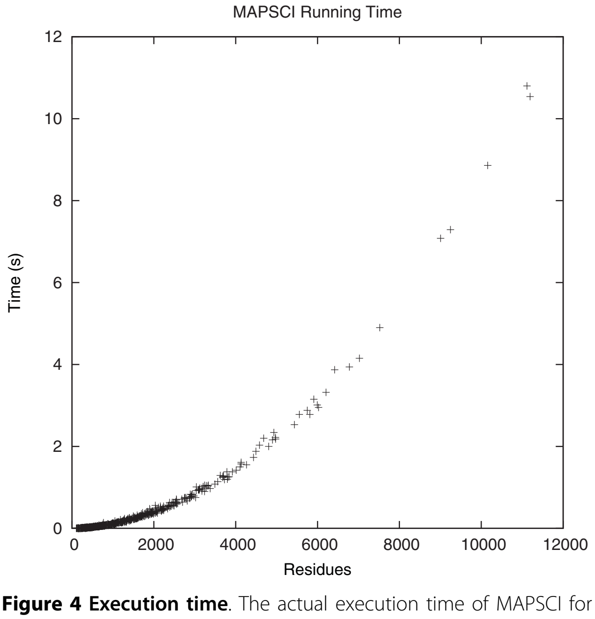


Figure 4 Execution time. The actual execution time of MAPSCI for all families in the benchmark datasets plotted in terms of the total number of residues per family.

Proof. For each *j*, we consider two distinct cases for J_j: either it is a coordinate triple, *x*, or a gap. If J_j is a gap, then the sum of the distances between $\overline{J}$ and each protein P_j along the *j*th column is $|I_n|\rho^2$, where ρ is the gap penalty. If J_j is a coordinate triple, *x*, then the sum of the distances between $\overline{J}$ and each protein P_j along the *j*th column is $|I_g|\rho^2 + \sum_{i\in I_n}||\vec{h}_{ij} - x||^2$, which is minimized, for $x = x_j = \frac{1}{|I_n|}\sum_{i\in I_n}\vec{h}_{ij}$. Therefore, if $|I_n|\rho^2 \geq |I_g|\rho^2 + \sum_{i\in I_n}||\vec{h}_{ij} - x_j||^2$, then the optimal choice for $\overline{J}_j$ is the coordinate triple x_j; otherwise, the optimal choice for $\overline{J}_j$ is a gap.

Computation of the optimal translation matrix

In this section, we show how to compute the optimal translation matrix T_i, for each *i*, for a given consensus structure $\overline{J}$. From Eq. (2), it is clear that the optimal T_i and T_j, for $i \neq j$ are independent of each other. Hence, in the following, we focus on the computation of T_i, for a specific *i*. The translation matrix T_i can be

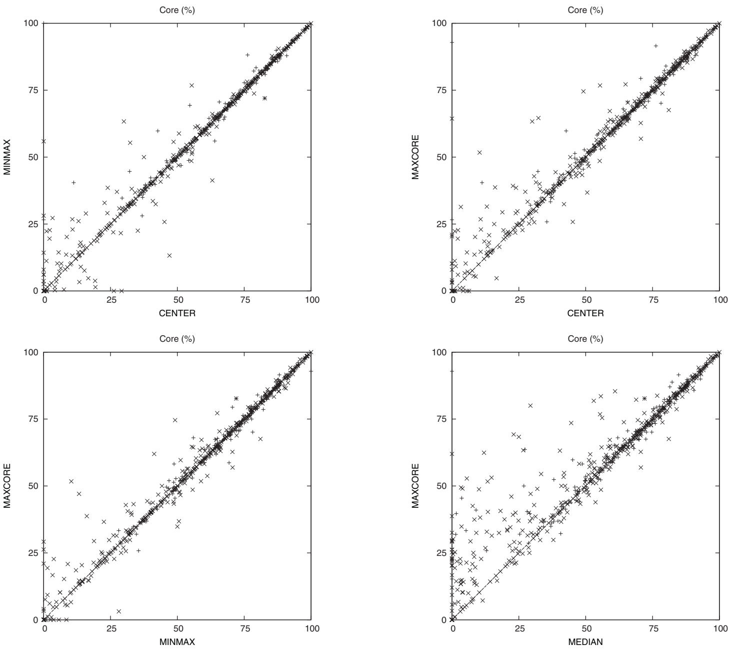


Figure 5 Consensus choice comparison. Comparison between the sizes of the aligned cores for different choices of initial consensus protein.

decomposed as $T_i = e \times t_i$, where $t_i \in R^{1 \times 3}$ 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 $\overline{J}$ and the *i*th protein P_i.

Let $A \in \mathbb{R}^{n \times 3}$ and $B \in \mathbb{R}^{n \times 3}$ consist of the coordinate triples from the consensus structure and the *i*th protein, respectively, after removing the mismatches. (Here n is the number of matches between the consensus structure and the *i*th protein, i.e., comparison of two non-gaps). Without loss of generality, assume $e^T A = [0, 0, 0]$, i.e., the coordinate triples in the consensus protein are centered at the origin. The optimal translation vector is the one that matches the centroids of the coordinate triple vectors from A and B as stated in the following theorem:

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* $t_i = \frac{1}{n} e^T B$.

More details can be found in [18].

Computation of the optimal rotation matrix

Next, consider the rotation matrix R_i. We can assume that the coordinate triple vectors from both A and B are centered at the origin. It follows that

$$\begin{aligned} S_i &= || A - BR_i ||^2 = \text{trace}\ ((A - BR_i)^T (A - BR_i)) \\ &= \text{trace}\ (A^T A) - 2\text{trace}\ (A^T BR_i) + \text{trace}\ (B^T B). \end{aligned}$$

Hence the minimum of S_i is obtained when trace $(A^T BR_i)$ is maximized.

Let the Singular Value Decomposition (SVD) [16] of $A^T B$ be $U \Sigma V^T$, where U and V are orthogonal and Σ is diagonal.

Theorem 3. *The optimal rotation matrix* R_i *that minimizes* $S_i = ||A - BR_i||^2$ *is given by* $R_i = UWV^T$, *where* $W = diag(1, 1, 1)$, *if* $det(UV^T) = 1$, *and* $W = diag(1, 1, -1)$, *if* $det(UV^T) = -1$.

More details can be found in [18].

Convergence of the algorithm

In this section, we show that MAPSCI converges, by showing that the SC-distance is non-increasing from one iteration to the next.

Recall that from Eq. (1),

$$SC = \sum_{1 \le j \le K} D(P_0, P_j) = \sum_{1 \le j \le K} || H_0 - (H_j - T_j) R_j ||_F^2.$$

Line 4 in MAPSCI decreases the distance between the consensus structure and each of the K proteins, since the dynamic programming produces an alignment with minimum cost. By the property of the center-star-like method, Line 6 leaves unchanged the distance between the consensus structure and each of the K proteins. By Theorems 2 and 3, the transformations computed in

Line 7 do not increase the distance between the consensus structure and the *j*th protein, for each *j*. It is clear that Line 8 does not change the pairwise distance, since the cost for aligning two gaps is zero. Finally, by Theorem 1, Line 9 does not increase the sum of the pairwise distances from the consensus structure to the other proteins. Hence, the SC-distance is non-increasing, and the algorithm converges.

Complexity analysis

Let n be the maximum length of the K proteins. Then the overall running time of the algorithm is $O(K^2n^2)$. (If we choose the initial consensus structure as the protein of median length, the running time is $O(Kn^2 + K^2n)$.) The run time analysis is similar to that of the algorithm in [5].

Conclusions

We have presented an algorithm, called MAPSCI, to compute a multiple structure alignment for a set of proteins, together with their consensus structure. The algorithm represents the input proteins and the consensus as sequences of coordinate triples and computes an approximation to the optimal multiple structure alignment that minimizes the sum of the pairwise distances between the consensus and each input protein. Experimental results on a benchmark datasets derived from the HOMSTRAD and SABmark databases show that the algorithm compares favorably with existing algorithms for multiple structure alignment (MAMMOTH and MATT).

Availability and requirements

- Project name: MAPSCI
- Project home page: `http://www.geom-comp.umn.edu/mapsci`
- Operating system(s): Platform-independent
- Programming language: C++
- License: Free BSD

Acknowledgements

Research of JY and RJ was sponsored, in part, by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAD19-01-2-0014, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

Adam Isom contributed to the implementation of the web-based tool.

The authors thank the MAMMOTH and MATT teams for sharing their source code.

Author details

[1]Department of Computer Science, Gettysburg College, Gettysburg, PA, USA. [2]Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, USA. [3]Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA.

Authors' contributions

JY contributed to the design of the algorithm and experiments, and drafting of the manuscript. II contributed to the experiments and the implementation of the algorithm. RJ contributed to the refinement of the algorithm and drafting of the manuscript. All authors read and approved the final manuscript.

Received: 24 August 2009
Accepted: 2 February 2010 Published: 2 February 2010

References

1. Guda C, Scheeff ED, Bourne PE, Shindyalov IN: A new algorithm for the alignment of multiple protein structures using Monte Carlo optimization. *Proceedings of the Pacific Symposium on Biocomputing: 3-7 January 2001; Hawaii* 2001, 275-286.
2. Lupyan D, Leo-Macias A, Ortiz AR: A new progressive-iterative algorithm for multiple structure alignment. *Bioinformatics* 2005, **21**:3255-3263.
3. Menke M, Berger B, Cowen L: Matt: Local Flexibility Aids Protein Multiple Structure Alignment. *PLoS Computational Biology* 2008, **4**:0088-0099.
4. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: The Protein Data Bank. *Nucleic Acids Research* 2000, **28**:235-242.
5. Ye J, Janardan R: Approximate multiple protein structure alignment using the Sum-of-Pairs distance. *Journal of Computational Biology* 2004, **11(5)**:986-1000.
6. Gusfield D: *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology* Cambridge University Press, Cambridge 1997.
7. Ye J, Janardan R, Liu S: Pairwise protein structure alignment based on an orientation-independent backbone representation. *Journal of Bioinformatics and Computational Biology* 2004, **2(4)**:699-717.
8. Waterhouse AM, Procter JB, A MDM, M C, Barton GJ: Jalview Version 2 - a multiple sequence alignment editor and analysis workbench. *Bioinformatics* 2009, **25(9)**:1189-1191.
9. Chemis3D: Molecular Viewer Applet. http://chemis.free.fr/mol3d/.
10. Mizuguchi K, Deane CM, Blundell TL, Overington JP: HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Science* 1998, **7**:2469-2471.
11. VanWalle I, Lasters I, Wyns L: SABmark - A benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics* 2005, **21**:1267-1268.
12. Madhusudhanm MS, Webb BM, Marti-Renom MA, Eswar N, Sali A: Alignment of multiple protein structures based on sequence and structure features. *Protein Engineering, Design & Selection* 2009, **22(9)**:569-574.
13. Venclovas C, Zemla A, Fidelis K, Moult J: Comparison of performance in successive CASP experiments. *Proteins* 2001, **45(S5)**:163-170.
14. Holm L, Sander C: Protein Structure Comparison by Alignment of Distance Matrices. *Journal of Molecular Biology* 1993, **233**:123-138.
15. Singh AP, Brutlag DL: Hierarchical protein structure superposition using both secondary structure and atomic representation. *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology: 21-26 June, 1997; Halkidiki* 1997, 284-293.
16. Golub GH, Van Loan CF: *Matrix Computations* Johns Hopkins University Press, Baltimore 1996.
17. Chew LP, Kedem K: Finding the consensus shape of a protein family. *Proceedings of the Eighteenth Annual ACM Symposium on Computational Geometry: 5-7 June 2002; Barcelona* 2002, 64-73.
18. Umeyama S: Least-square estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1991, **13(4)**:376-380.

doi:10.1186/1471-2105-11-71
Cite this article as: Ilinkin *et al.*: Multiple structure alignment and consensus identification for proteins. *BMC Bioinformatics* 2010 **11**:71.