MASS: multiple structural alignment by secondary structures

O. Dror¹,*, H. Benyamini², R. Nussinov²,³ and H. Wolfson¹

¹School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel, ²Sackler Institute of Molecular Medicine, Sackler Faculty of Medicine, Tel Aviv University, Tel Aviv 69978, Israel and ³Basic Research Program, SAIC-Frederick, Inc, Laboratory of Experimental and Computational Biology, NCI - FCRDC, Bldg 469, Rm 151, Frederick, MD 21702, USA

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT
We present a novel method for multiple alignment of protein structures and detection of structural motifs. To date, only a few methods are available for addressing this task. Most of them are based on a series of pairwise comparisons. In contrast, MASS (Multiple Alignment by Secondary Structures) considers all the given structures at the same time. Exploiting the secondary structure representation aids in filtering out noisy results and in making the method highly efficient and robust. MASS disregards the sequence order of the secondary structure elements. Thus, it can find non-sequential and even non-topological structural motifs. An important novel feature of MASS is subset alignment detection: It does not require that all the input molecules be aligned. Rather, MASS is capable of detecting structural motifs shared only by a subset of the molecules. Given its high efficiency and capability of detecting subset alignments, MASS is suitable for a broad range of challenging applications: It can handle large-scale protein ensembles (on the order of tens) that may be heterogeneous, noisy, topologically unrelated and contain structures of low resolution.

Keywords: Structural Bioinformatics, Subset alignment, Non-sequential alignment, Non-topological motif, Supersecondary structural motif

Availability and supplementary information: MASS program and more information about it are available on http://bioinfo3d.cs.tau.ac.il/MASS

Contact: oranit@post.tau.ac.il

INTRODUCTION
One of the most important concepts in molecular biology is that the function of a protein is determined by its 3D structure (Branden and Tooze, 1999). Thus, proteins sharing a common substructure may have a similar function. This observation gave rise to the development of structural alignment tools. Such tools are becoming extremely important with the rapidly increasing number of known protein structures, the outcome of the Structural Genomics project.

Many algorithms have been developed to solve the pairwise structural alignment task. For good reviews see (Brown et al., 1996; Lemmen and Lengauer, 2000; Eidhammer et al., 2001). In contrast, only a few methods are available for aligning multiple structures. However, it is obvious that a multiple alignment carries significantly more information than a pairwise one. Multiple structural alignment is a much more powerful tool for: (i) classification of existing and newly discovered proteins; (ii) gaining insights into evolutionary relations between proteins; (iii) detecting motifs common to a group of proteins that share a certain function or binding property; and (iv) structure prediction algorithms like homology modeling and threading.

Most of the methods for aligning multiple molecules find common substructures through a series of comparisons between pairs of molecules (Eidhammer et al., 2001). The best known methods are SSAPm (Taylor et al., 1994), Prism (Yang and Honig, 2000b), STAMP (Russell and Barton, 1992), (Gerstein and Levitt, 1996) and (Akutsu and Sim, 1999). Such methods have an inherent limitation since in each pairwise comparison the only available information is about the two molecules involved. Hence, alignments optimal for the whole set can be disregarded, if they are not also optimal for every pair (Eidhammer et al., 2001). Additionally, the majority of these methods use dynamic programming (Needleman and Wunsch, 1970). As a result they have the disadvantage of being dependent on the sequence order of the polypeptide chain. Our method, MASS, is a truly multiple-based approach. It considers all the given molecules simultaneously, rather than initiating from pairwise comparisons. Three other multiple-based
methods are (Escalier et al., 1988), MUSTA (Leibowitz et al., 2001) and MultiProt (Shatsky et al., 2002).

MASS is capable of detecting subset alignments. It can find not only structural cores common to the whole given set of molecules but also cores shared by non-predefined subsets. This makes the method insensitive to the presence of structurally dissimilar molecules and may aid in distinguishing between subsets of similar and dissimilar molecule structures.

Another important characteristic of our method is its use of secondary structure elements (SSEs). The rationale for this approach is that SSEs are inherent components of protein structures, providing a stabilizing scaffold. Consequently, SSEs are evolutionarily highly conserved while mutations frequently occur at flexible loops.

The benefits that are gained by using SSEs are: (i) Efficiency. The average number of SSEs in a protein is about 15, that is, more than an order of magnitude smaller than the average number of residues in a protein (∼300 residues). Representing the proteins by their SSEs introduces great savings in structural description and as a result significantly reduces the time complexity; and (ii) Accuracy and noise filtering. Globular proteins are dense molecules. Hence, any pair of arbitrary proteins can be superimposed in a way that many of their atoms are matched. An SSE-based superposition algorithm filters out beforehand the noisy alignments and alleviates this problem. Representing protein structures by their SSEs has been successfully used in several methods for pairwise alignment (Koch et al., 1996; Alesker et al., 1996; Alexandrov and Fischer, 1996; Grindley et al., 1993; Lu, 2000; Yang and Honig, 2000a). MASS disregards the order of the SSEs along the polypeptide chain and therefore can find non-sequential and even non-topological alignments.

MASS was applied to various protein ensembles. The novel combination of being SSE-based, highly efficient, and capable of detecting non-topological and subset structural similarities is demonstrated below.

PROBLEM DEFINITION

The pairwise case: We are given two proteins. Each protein is represented as a set of points in 3D space, where each point is associated with an atom’s position. The task is to find the largest set that is congruent to two subsets of points, one from each input set (i.e. protein). In computational geometry this problem is known as the largest common point set (LCP) problem (Alt and Guibas, 1999).

The best alignment between the two proteins is the rigid transformation which superimposes their two largest congruent subsets. Specifically, if \( P = \{ p_i \} \) and \( Q = \{ q_j \} \) are the two largest congruent subsets of the input sets, then the best alignment between the input sets is defined by a rigid transformation \( T \), which satisfies \( P = T(Q) = R(Q) + t \) where \( R \) is a \( 3 \times 3 \) rotation matrix and \( t \) is a translation vector. The core of the alignment, also termed match list or correspondence list, is defined as \( \{(p_i, q_j): p_i = T(q_j)\} \).

The above formulation of the problem is a theoretical one. It assumes noise-free data, i.e. that all atom positions are known exactly. In addition, it requires an exact alignment between the matched atoms (i.e. \( P = T(Q) \)). However, in practice, it may not be possible to find an exact alignment. Therefore, a more practical task is to find two subsets \( P \) and \( Q \) that are \( \epsilon \)-congruent. Specifically, for a predefined \( \epsilon > 0 \), the goal is to find the two largest subsets of the input sets, \( P \) and \( Q \), and a rigid transformation, \( T \), so that \( \text{distance}(P, T(Q)) < \epsilon \).

The multiple case: This problem is a variant of the multiple-LCP problem i.e. given a collection of \( m \) point sets, the task is to find the largest set of points, of which an \( \epsilon \)-congruent copy appears in each of the input sets. Unfortunately, the multiple-LCP problem turns out to be NP-hard, when the number of input sets is unbounded. In fact, it is even hard to find an approximate solution, i.e. finding a common point-set which is within a factor of \( f(n) \) of the largest common point-set, where \( n \) is the size of the smallest point-set (Akutsu and Halldorsson, 2000).

In practice, we are interested in solving an even more complicated problem. Specifically, a biologically interesting solution might not necessarily be the largest common substructures. Thus, we would like to detect smaller common substructures as well, as long as they are larger than a predefined threshold.

The multiple subset case: The problem is further complicated by the requirement to find solutions, where only a subset of the input proteins is well aligned. This requirement complicates the problem since the number of subsets is exponential in the number of the input molecules. Additionally, the definition of the requirement is quite vague mathematically, since there is a trade-off between the size of the subset and the size of its core. Therefore, we use a scoring function to find the best multiple alignments. The score of each alignment is based on the core size \( l \) and the number of proteins that participate in the alignment \( k \). We have used the following scoring function: \( f(l, k) = k \cdot \binom{l}{k} \).
Here, we propose a heuristic algorithm to the hard problem stated above that runs in polynomial time and yields good results.

**METHOD**

*Input:* The input for the algorithm is a set of $m$ proteins: $P_1, P_2, \ldots, P_m$. For each protein there are two inputs: One is the sequence of the 3D coordinates of its atoms in PDB format (Bernstein et al., 1977). The other is the assignment of SSE types to each residue in one of the following formats: PDB (Bernstein et al., 1977), DSSP (Kabsch and Sander, 1983) and DSSCont (Andersen et al., 2002).

**General strategy**

The algorithm is based on the assumption that a multiple alignment is a good one only if its core contains at least two SSEs. So, as a first step toward finding good multiple matches we detect ordered pairs of SSEs, termed *bases* whose $\epsilon$-congruent copies appear in several proteins.

A set of $\epsilon$-congruent bases, each taken from a different protein, induces a *multiple base alignment* between the respective proteins. Specifically, assume that $\{b_1, b_2, \ldots, b_k\}$ is a set of $\epsilon$-congruent bases, taken from proteins $P_{i_1}, P_{i_2}, \ldots, P_{i_k}$ respectively. We pick the first base ($b_1$) as a pivot and compute all the $k-1$ rigid transformations between this base and each of the remaining ones. The resulting $k-1$ dimensional vector of rigid transformations, $(T_{12}, T_{13}, \ldots, T_{1k})$ defines a multiple alignment between $P_{i_1}, P_{i_2}, \ldots, P_{i_k}$. The core of this alignment contains at least two SSEs, but it can be extended into a larger substructure.

Suppose now that proteins $P_{i_1}, P_{i_2}, \ldots, P_{i_k}$ can be aligned so that the core of their alignment contains more than one base. In such a case, we will get several redundant multiple base alignments i.e. alignments with almost the same transformations (one alignment for each base in the core). Therefore, in the next stage of the algorithm we cluster the initial multiple base alignments. The alignments in each cluster are merged together so that the core of the new alignment is the union of the cores of the original alignments. As a result, we get a smaller non-redundant set of *global multiple alignments*.

Next, we extend the clustered alignments by finding additional matching residues, which do not necessarily belong to SSEs. A score is given to each alignment according to the core size and the number of aligned proteins. Finally, the highest scoring alignments are reported.

**Detailed description**

Representation of secondary structure elements: Each SSE (a helix or a strand) is represented by its axis. The axis is a directed line segment where (i) its line is the least squares line from all the $C_\alpha$ atoms; (ii) its length is determined by the projection of the terminal $C_\alpha$ atoms; and (iii) its direction is from the N-terminus to the C-terminus (see Fig. 1).

Detection of multiple base alignments: In this stage we use Geometric Hashing (Nussinov and Wolfson, 1991) to efficiently detect bases, whose $\epsilon$-congruent copies appear in several proteins. Each base is represented by a *fingerprint* which is invariant to a 3D rigid transformation. The fingerprint is composed of: (i) the types of the two SSEs (i.e. helix or strand); (ii) the angle between their axial vectors; (iii) the midpoint-to-midpoint distance between their axes; and (iv) their line distance, which is the closest distance in space between their least-squares lines (see Fig. 2).

The bases are inserted into a 5D grid addressed by their fingerprints. The resolution of the grid is determined by the tolerance that we allow between two similar fingerprints. $\epsilon$-congruent bases are represented by a similar fingerprint. Therefore, they reside close enough to each other in the grid. However, note that they do not necessarily have...
to reside together in the same grid voxel. Then, for each grid voxel, we retrieve all the bases in that voxel and in the adjacent voxels and group them into the same Base Bucket (BB) (see Fig. 3). The bases in a base bucket, are stored in columns, according to the protein they belong to. The columns are sorted in an increasing order according to the proteins’ indices. This arbitrary order is necessary for the clustering in the next stage.

A set of bases, each taken from a different column of the same BB, defines a multiple base alignment between the respective proteins. The number of multiple base alignments defined by a BB is exponential in the number of proteins it stores. To see this, let us define \( BB(i) \) as the set of all bases, stored in column \( i \). Using this notation, we can write the BB as the set of its columns: \( BB = \{ BB(1), BB(2), \ldots, BB(s) \} \).

The number of choices equals to \( \sum_{(BB(i_1), \ldots, BB(i_k)) \in 2^{BB}} |BB(i_1)| \cdots |BB(i_k)| \), where \( 2^{BB} \) is the power set of the BB. In the following, we will propose heuristics for choosing in polynomial time the best alignments from this exponential number of candidates.

The multiple alignment between \( k \) \( \epsilon \)-congruent bases is defined by a \( k \times 1 \) dimensional vector of rigid transformations, \( (T_{i_1}, T_{i_2}, \ldots, T_{i_k}) \) where \( T_{i_l} \) is a transformation between the pivot base \( b_{i_1} \) and \( b_i \) (\( 2 \leq i \leq k \)). We can conclude that for a base bucket BB the set \( \mathcal{S} = \bigcup_i \bigcup_{i < j} BB(i) \times BB(j) \) of pairwise base alignments defines all its multiple base alignments. In order to compute \( \mathcal{S} \) we need to uniquely define the alignment between two \( \epsilon \)-congruent bases: \( b_1 = (S_1^1, S_2^1) \) and \( b_2 = (S_1^2, S_2^2) \).

There are several ways to do so. One of the ways is as follows: we represent each SSE \( (S_i^j) \) by the list of its \( C_\alpha \) atoms and simultaneously enumerate all possible ways of aligning the atom list of \( S_1^1 \) with that of \( S_2^1 \) and the atom list of \( S_1^2 \) with that of \( S_2^2 \). Note that the atoms of \( S_1^1 \) and \( S_2^1 \) need to be aligned consecutively. Therefore, if \( k \) is the number of \( C_\alpha \) atoms in each SSE, then there are \( O(k) \) possibilities (shifts) for matching the atom lists of \( S_1^1 \) and \( S_2^1 \) and thus \( O(k^2) \) possible match lists between the two pairs of SSEs. For each possible match list we use the Least-Squares Fitting approach (Kabsch, 1978) to compute in \( O(k) \) time the transformation that optimally aligns the matched atoms. The best alignment between the bases is the one that minimizes the RMSD. The total run time is \( O(k^3) \). Another way to align \( b_1 \) and \( b_2 \) is to uniquely define an axis system on each base and superimpose the two axis systems. This alignment is useful especially when information about the protein atoms is not supplied and the only available information is about their secondary structures (e.g. information extracted from EM density maps (Chiu et al., 2002) or models).

Clustering: For a pair of proteins that have more than one base in common, we may get redundant base alignments, i.e. alignments with almost the same transformation, but with a different local core. Therefore, for each pair of proteins, we cluster the base alignments to find the ones with approximately the same transformation. After the clustering, alignments with almost the same transformation are merged into a new alignment. The match list of the new alignment is the union of the original match lists and its transformation is the one that aligns the atoms of the new match list with the smallest RMSD (Kabsch, 1978). The clustering method that we use is partially similar to the one introduced in (Rarey et al., 1996). This algorithm works in an iterative manner. At the beginning, every pairwise alignment is considered as a separate cluster, represented by its transformation. The distance between two clusters is defined as the RMSD between the images of their transformations applied on an atom set, taken from the pivot protein. As long as the minimum distance between two clusters is below a predefined threshold, the two closest clusters are merged into a single cluster. The transformation of the new cluster is defined as the transformation of one of the original clusters. In a similar manner to the complexity analysis in (Rarey et al., 1996) it can be shown that the complexity is \( O(p^2 \log p) \) where \( p \) is the number of alignments to cluster.

Residue extension: When reaching this stage, each pairwise alignment is associated with a transformation that optimally superimposes a set of SSEs from the second protein onto a similar set from the pivot protein. Now
Computing the best global multiple alignments: In this stage we return to the base buckets. Our aim is to detect the multiple alignments with the largest cores. As noted before, this is an NP-hard problem. Therefore, we provide only a heuristic solution: For each BB we compute the set of best multiple alignments in a recursive manner over its columns. The recursion works as follows: in each stage we have a set of multiple base alignments, obtained by the previous stage of the recursion. For each such alignment, \((b_1, \ldots , b_k)\), we check if there is a base, \(b_{k+1}\), from the current column that, when added to the alignment, will improve its score. The score of the multiple alignment is defined as \(k \cdot \binom{l}{2}\) where \(k\) is number of aligned proteins and \(l\) is the core size. The core of the multiple alignment, \(\text{Core}(b_1, \ldots , b_{k+1})\), is \(\text{Core}(b_1, \ldots , b_k) \cap \text{Core}(b_1, b_{k+1})\) (note that we have the cores of the pairwise alignments from the previous stages).

The returned set of multiple alignments at the end of this stage also contains trivial alignments, each with one base, for each base in the column. This means that each of the bases in the current column will also act as a pivot in a multiple alignment for the previous recursion calls.

**Complexity**

Due to the lack of space, we cannot go into details and we will only analyze the worst case complexity. In practice, the complexity is much smaller. In the following, (i) \(m\) is the number of proteins; (ii) \(k\) is the number of residues in an SSE; and (iii) \(s\) and \(n\) are the number of SSEs and the number of residues found in each protein respectively. In a typical protein \(n \sim 300\), \(k \sim 10\) and \(s \sim 15\).

The number of bases for each protein is \(O(s^2)\). Thus, in the worst case, for each pair of proteins we construct, cluster and extend \(O(s^4)\) pairwise alignments. This results in \(O(m^2(s^4k^3 + s^8 \log s + s^4n))\) time where \(O(m^2)\) is the number of ways of pairing two proteins. In practice, the complexity is much smaller, since we only construct the pairwise alignments defined by the BBs and the clustering reduces their number even more.

The number of evaluated multiple alignments is linear in the number of bases, since each base can be a pivot for only one multiple alignment. After the pivot of a multiple alignment is defined, i.e. a trivial multiple alignment is constructed, we need to look at every base from the rest of the BB columns and check its intersection with the alignment in \(O(n)\) time. Since we have \(O(m^2s^2)\), it takes \(O(m^2s^2n)\) time to construct a single multiple alignment and \(O(m^2s^4n)\) time to construct all of them. Finally, the time complexity for the whole algorithm is bounded by \(O(m^2s^4(k^3 + s^2 \log s + n))\). However, according to the experimental results the actual running times are significantly lower.

**RESULTS**

We have applied MASS to several protein ensembles (PDB codes are listed in Table 2). All the experiments were performed on a PC workstation (Pentium© 4 1800 MHz processor with 1GB RAM). Following is a description of the results, demonstrating the power of MASS in addressing various challenging cases of structural alignment. Examples include: (i) ensembles of proteins with different topology; (ii) ensembles in which only a part of the molecules share a common structural motif (subset alignment); and (iii) ensembles with large number (on the order of tens) of proteins.

**Non-topological motif detection**

In the following two examples, the ensembles share a common SSE motif, in spite of having a different topology. In topological motifs, the order and the direction of the corresponding SSEs along the polypeptide chain are conserved while in non-topological they are not.

**Helix bundle ensemble:** The ten proteins in this ensemble belong to four different folds and six different superfamilies in the SCOP database (Murzin et al., 1995). The running time of MASS on this ensemble was 48 seconds. Figure 4a presents their structural alignment. 29 residues were structurally aligned within an RMSD of 2.4Å. MASS detected four conserved helices that form a bundle. Figures 4b and 4c show that the alignment is non-topological. This ensemble was also aligned by MUSTA (Leibowitz et al., 2001). MASS detected two additional conserved α-helices. A possible explanation for this larger motif detection is the fact that MASS is secondary structure oriented and thus is directed to find solutions that contain more SSEs.

**C2 domain ensemble:** This ensemble consists of ten proteins taken from the two families of the ‘C2 domain’ superfamily (each protein belongs to a different domain) (Murzin et al., 1995). The running time was 32 seconds. Figure 5a shows the core of the alignment. It consists of 58 residues within an RMSD of 1.9 Å and forms a sandwich of eight β-strands (one strand is not conserved in 3pbA and 1bdyA). Figures 5b and 5c show that the alignment...
Fig. 4. Helix bundle ensemble: (a) The structural alignment of all ten proteins. The backbone of the proteins is colored in gray. Their common core is shown by assigning a different color to each conserved helix. (b) The alignment of the core residues. Secondary structure assignments, determined by the DSSP program, are shown in brackets next to the residue index (H stands for a helix). Residues of matched secondary structure regions are colored in gray and yellow alternately. Note that only a small loop of two residues is located between H4 (residues: 84–91) and H5 (residues: 93–105) of 256bA protein and between H2(53–60) and H3(63–70) of 3inkC protein. (c) The schematic TOPS representation (Flores et al., 1994). Triangles represent strands and circles helices. Corresponding secondary structure regions are drawn in the same color. As one can see the solution is non-topological.

is non-topological. In fact, there are two topological groups. The first consists of proteins from the ‘PLC-like (P variant)’ family and the second consists of proteins from the ‘Synaptotagmin-like (S variant)’ family. The two families are related by circular permutation (Murzin et al., 1995).

For both ensembles, MASS detected conserved non-topological motifs. These could not have been detected neither by multiple sequence alignment, nor by multiple structural alignment which is based on dynamic programming.

Subset alignment detection

We applied MASS to an ensemble of eleven proteins belonging to five different families of the ‘PLP-dependent transferases’ superfamily. The running time was 1 m 21 s. All eleven proteins were aligned with a core containing 89 residues within an RMSD of 2.1 Å. Additionally, although it is considered very hard to align all the seven strands of the sheet, especially the two external β-strands (Kack et al., 1999), MASS detected a core containing the complete seven-stranded β-sheet of the fold and three α-helices (Fig. 6a).
Fig. 5. C2 domain ensemble: (a) The core of the alignment of all ten proteins. The backbone of the proteins is colored in gray and only part of it is displayed. The common core is shown by assigning a different color to each conserved β-strand. (b) The match between the eight conserved strands. The forth strand (blue) is not conserved in 3rpbA and 1bdyA proteins (c) The TOPS diagrams (Flores et al., 1994) demonstrate that the alignment is non-topological.

Moreover, a subset alignment of ten proteins has been detected. The core of the alignment contains 100 residues with an RMSD of 2.1 Å. This core has an additional α-helix (see Fig. 6b). Protein 1bt4 is an exception lacking this extra helix. This subset alignment agrees with the evolutionary relationship tree that has been generated by all-against-all pairwise comparisons (Kack et al., 1999).

Large-scale structural alignments

MASS can be applied on the order of tens of proteins in practical running times on a standard PC. To demonstrate the efficiency of MASS on such large sets of proteins, we have applied it to the following three SCOP ensembles (Murzin et al., 1995): (i) Serine proteases—all structures from the ‘Prokaryotic trypsin-like serine protease’ SCOP family (68 molecules); (ii) PK beta barrel—all structures from the ‘Pyruvate kinase beta-barrel domain’ SCOP family (66 molecules); and (iii) unrelated proteins—a compiled set of 80 proteins, where each protein was taken from a different SCOP fold. It contains proteins from the four major SCOP classes: all-α, all-β, α+β and α/β.

The details of the ensembles and the running times are summarized in Table 1. The results show that the running time is influenced by: (i) the number of molecules; (ii) the average molecular size (and the average number of SSEs in a molecule); and (iii) the structural variance among the molecules. The first two parameters are expected and indeed increase the running time as they grow (see complex-
ity analysis in the Method section). For example, the average molecular size in the ‘serine protease’ ensemble is 243 residues versus 98 in the ‘PK beta-barrel’ ensemble. The running times were 1 h 25 m and 1 m 28 s respectively. We attribute the difference in the running times to the difference in the molecular size, since the number of molecules in both ensembles is almost the same (68 and 66). Structural variance also influences the running time. The more structurally variable is the ensemble, the shorter the running time is. This is shown by comparing the ‘serine protease’ ensemble to the compiled set of unrelated proteins, which represents a much more structurally variable set. The running times were 1 h 25 m and 22 m respectively. This difference in the running times is attributed to the difference in the structural variance within each ensemble, since the average molecular size and the number of molecules in the unrelated set are higher than those of the serine proteases (300 and 80 versus 243 and 68). The reason for the influence of structural variance on the running time is that the more structurally homogeneous is the input, more SSE bases are stored in the same grid voxel.

CONCLUSIONS AND FUTURE WORK

We have described a novel method for aligning multiple protein structures and detecting their spatial core. The experimental results have shown that MASS is capable of detecting not only structural cores common to the whole set of proteins, but also cores common only to subsets of it. This novel utility is extremely important in ‘real-life’ structural alignment practice, where subset similarity may exist and is not known a priori.

MASS exploits a secondary structure representation of proteins. By this, many noisy solutions are filtered out. This further makes the method highly efficient and capable of aligning tens of protein molecules. The method disregards the sequence order of secondary structure elements along the polypeptide chain. Thus, it can find non-sequential and even non-topological structural motifs. This is a clear advantage over sequence alignment, or structural alignment that is based on dynamic-programming. MASS program can be run in two alternative modes: (i) using SSE information only; and (ii) using both SSE and atomic information, where the algorithm fine-tunes the transformations at the atomic level. The first mode can be used for cases where only SSE information exists. Additionally, in this mode the complexity and the running time are further reduced. Therefore, it may be useful for large-scale datasets or for cases that contain only SSE data, for example, theoretical models and low resolution structures. One novel and potentially very useful application is likely to be in the structure determination of large proteins or of super-molecular assemblies from cryo-electron microscopy (EM). While it is not possible to obtain the positions of single residues, the EM density maps may allow assignment of secondary structure elements (Chiu et al., 2002). Considering the noise inherent to such density maps, comparing them to PDB folds in their secondary structure representation may provide a powerful tool for structural interpretation of EM data.
### Table 1. Multiple structural alignment of large-scale datasets

<table>
<thead>
<tr>
<th>Ensemble Name</th>
<th>No. of Mol.</th>
<th>Avg. Mol. Size</th>
<th>Avg. No. of SSEs</th>
<th>Run Time (h:mm:ss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serine proteases</td>
<td>68</td>
<td>243</td>
<td>13</td>
<td>1:25:03</td>
</tr>
<tr>
<td>PK beta-barrel</td>
<td>66</td>
<td>98</td>
<td>4</td>
<td>0:01:28</td>
</tr>
<tr>
<td>unrelated proteins</td>
<td>80</td>
<td>300</td>
<td>14</td>
<td>0:22:01</td>
</tr>
</tbody>
</table>

The first ensemble consists of all proteins of the prokaryotic trypsin-like serine proteases. The second consists of all proteins from the PK beta barrel family. In contrast, the third ensemble is a compiled set with a high degree of structural variance (see text for details). It can be seen that the running time is influenced by the average molecular size, the average number of SSEs in a molecule and the structural variance among the proteins of the ensemble.

### Table 2. Data Set: The first four letters of a protein name are the PDB code, followed by chain id and the residue numbers for the first and the last residue

<table>
<thead>
<tr>
<th>Ensemble Name</th>
<th>Proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix bundle</td>
<td>1flx, 1aqp, 1bbhA, 1bgeB, 1le2, 1cbb, 256BA, 2cycyA, 2hmzA, 3mkC</td>
</tr>
<tr>
<td>C2 domain</td>
<td>1d5sA, 1esyA, 1a25A, 1bdyA, 1dsyA, 1gmiA, 1isqA, 1lqw, 1tsy, 3pbaA</td>
</tr>
<tr>
<td>PLP-dependent transferases</td>
<td>2plpA, 1ax4A, 4gssA, 1oaoA, 2dkk, 1gq5A, 1bd, 1ordA, 1ay4A, 1awa, 1ic2A</td>
</tr>
<tr>
<td>Serine proteinase</td>
<td>1arc, 1arb, 1qrwA, 1t1q4A, 1t1q5A, 1t1q6A, 1t1q7A, 1t1q8A, 1t1q9A, 1t1qaA</td>
</tr>
<tr>
<td>PK beta-barrel</td>
<td>1a49A:116-217, 1a49B:716-817, 1a49C:1316-1417, 1a49D:1916-2017, 1a49E:3116-3217, 1a49F:3716-3817</td>
</tr>
<tr>
<td>unrelated proteins</td>
<td>1ad3A, 1a2g, 1h7, 1a1o, 1aorA:211-605, 1arb, 1at0, 1bdks, 1bqv, 1c82A, 1cby, 1lcjy:142-721, 1lc5A, 1lca, 1csh, 1dsiA, 1dhs, 1ndpa:201-469, 1ds9A, 1ldyA, 1lid4A, 1leu1A:4-625, 1lewqa:267-541, 1lh0A, 1lh2A:284-583, 1lheh:210-574, 1lg9A, 1lgof:151-537, 1igrA, 1ibbnA:270-549, 1ihec, 1i7wa, 1iijaA, 1ijetA, 1jlsA, 1jlsA1, 1k30A, 1k8hA, 1knb, 1kypA:825-938, 1kypA, 1lka:110-379, 1llp, 1lxx, 1lmt, 1lns, 1loxp, 1o35A, 1oxexA, 1oxa, 1paq, 1pqqa, 1psg, 1psylA:111-421, 1psl, 1nml, 1nqtb, 1nvy, 1xaxa, 1xax, 1xyg, 2a8k, 2aak, 2act, 2bno, 2bpa1, 2dni, 2enp, 2e1, 2pgd:177-473, 2pl, 2vik, 4aahA, 4bcl, 7ataA, 7acn:2-528</td>
</tr>
</tbody>
</table>

### ACKNOWLEDGEMENTS

This research has been supported in part by the ‘Center of Excellence in Geometry Computing and its Applications’ funded by the Israel Science Foundation (administered by the Israel Academy of Sciences) and by grants of Tel-Aviv University Basic Research Foundation. The research of H.J. Wolfson and O. Dror has been partially supported by the Hermann Minkowski-Minerva Center for Geometry at Tel Aviv University. The research of H. Benyamini has been supported by the Eshkol Fellowship funded by the Israeli Ministry of Science. The research of R. Nussinov has been funded in whole or in part with Federal funds from the National Cancer Institute, National Institutes of Health, under contract number NO1-CO-12400. The content of this publication does not necessarily reflect the view or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organization imply endorsement by the U.S. Government.

### REFERENCES


