BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 11 2006, pages 1343–1352
doi:10.1093/bioinformatics/btl098

Structural bioinformatics

A permissive secondary structure-guided superposition tool for clustering of protein fragments toward protein structure prediction via fragment assembly

Gilad Wainreb[1], Nurit Haspel[2], Haim J. Wolfson[2] and Ruth Nussinov[1,3,*]

[1]Sackler Institute of Molecular Medicine, Department of Human Genetics, Sackler Faculty of Medicine,
[2]School of Computer Science, Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel and
[3]Basic Research Program, SAIC-Frederick, Inc., Laboratory of Experimental and Computational Biology,
NCI-Frederick, Building 469, Rm 151, Frederick, MD 21702, USA

Received on July 31, 2005; revised on February 16, 2006; accepted on March 11, 2006
Advance Access publication March 16, 2006
Associate Editor: Martin Bishop

ABSTRACT

Motivation: Secondary-Structure Guided Superposition tool (SSGS) is a permissive secondary structure-based algorithm for matching of protein structures and in particular their fragments. The algorithm was developed towards protein structure prediction via fragment assembly.

Results: In a fragment-based structural prediction scheme, a protein sequence is cut into building blocks (BBs). The BBs are assembled to predict their relative 3D arrangement. Finally, the assemblies are refined. To implement this prediction scheme, a clustered structural library representing sequence patterns for protein fragments is essential. To create a library, BBs generated by cutting proteins from the PDB are compared and structurally similar BBs are clustered. To allow structural comparison and clustering of the BBs, which are often relatively short with flexible loops, we have devised SSGS. SSGS maintains high similarity between cluster members and is highly efficient. When it comes to comparing BBs for clustering purposes, the algorithm obtains better results than other, non-secondary structure guided protein superimposition algorithms.

Availability: SSGS is available for download at http://www.cs.tau.ac.il/~wainreb

Contact: ruthn@ncifcrf.gov

Supplementary information: Supplementary data are available at *Bioinformatics* online.

INTRODUCTION

For over a quarter of a century (Karplus, *et al.*, 1997), ideas on protein folding have been dominated by two interrelated concepts: the Levinthal paradox (Levinthal, 1968) and a necessity for folding intermediates (Tsong *et al.*, 1972). Levinthal argued that, because there is an astronomical number of conformations open to the denatured state of a protein, an unbiased search through these would take ‘an eternity’. The early 1990s witnessed a revolution of the concepts (Bryngelson and Wolynes, 1989). The problem of protein folding was viewed in the context of populations and ensembles. The ‘folding funnel’ model argued that instead of following a single pathway, the population may take various routes in a free-energy funnel landscape (Baldwin, 1994, 1995; Bryngelson, 1989; Dill, 1997; Karplus *et al.*, 1995; Karplus and Shakhnovitch).

The roughness of the funnel slopes (Bryngelson *et al.*, 1995; Onuchic *et al.*, 1996) reflects the inability to energetically satisfy all the interactions in any given conformation (Onuchic *et al.*, 1995) and the transition state ensembles may be conformationally restricted (Martinez *et al.*, 1998). Although the free energy funnel is described as having one global minimum conformation, at the bottom of the funnel is an ensemble, with many conformers likely to be important for biological function. The hydrophobic collapse model (Dill *et al.*, 1995; Lesk and Rose, 1981; Pace *et al.*, 1996; Yue *et al.*, 1995) predicts that a protein will rapidly collapse (Ben-Naim, 1980; Bryngelson, 1989; Dill, 1990; Struthers *et al.*, 1996) because of its hydrophobic side chains, invoking interactions such as van der Waals and electrostatic. The hierarchical model postulates that the unit from which a fold is constructed is the outcome of a combinatorial assembly of a set of folding units. The assemblies associate to form intra-molecular domains. The hydrophobic folding units (HFUs) possess relatively strong hydrophobic cores, and their hydrophobic interactions with their surroundings, or with other units, are weaker. They are compact and may consist of non-contiguous segments on the amino acid chain (Struthers *et al.*, 1996). According to the building block (BB) model, an HFU consists of contiguous segments of the chain defined as building blocks (BBs). If a BB is removed from the chain, the most highly populated conformation of the extracted peptide in solution would very likely be similar to that of the BB when it is embedded in the native protein structure, even though an alternative conformation may be selected in the combinatorial assembly. Tsai *et al.* (Tsai *et al.*, 2000; Tsai and Nussinov, 2001) devised an empirical fragment-size-independent scoring function that measures the relative conformational stability of protein fragments and favors folding units that are compact, isolated and highly hydrophobic modules.

[*]To whom correspondence should be addressed.

We have devised a three-stage protein prediction scheme which follows the protein folding process. The sequence is first cut into structurally assigned BBs (Haspel *et al*., 2003b). Next, we perform a combinatorial assembly to predict the BBs relative 3D arrangement (Inbar *et al*., 2005; Tsai *et al*., 2004). In the third stage, we refine and rank the assemblies. To implement the first stage of the prediction scheme and identify BBs in a given protein sequence there is a need for a BB structural template library. In order to relate each BB to a characteristic profile that represents its typical sequence, we clustered BBs from a BB library derived by cutting proteins from the PDB. The BB library consisted of ~68 000 protein fragments, exhibiting a relatively high conformational stability as measured by the scoring function (Tsai *et al*., 2000; Tsai and Nussinov, 2001). They were assembled from a structurally non-redundant protein dataset. To relate a BB with a structural template, the clusters that emerge from the clustering process should have enough members to allow detection of a recurring sequence pattern. To achieve this goal while maintaining high similarity between BB cluster members, we used BBs both off and on the main folding route and developed a permissive novel pairwise superimposition tool, coined Secondary Structure Guided Superimposition tool (SSGS) to efficiently detect structural similarity among the BBs and cluster them. It aligns Cα atoms of two proteins while considering their secondary structure assignment, allowing a more permissive alignment of loop regions. The algorithm combines pose clustering and dynamic programming (DP) to find the best order-dependent global superposition. The DP aims to achieve an alignment that satisfies both affine mismatch and affine gap characteristics; namely, an alignment that favors fewer larger gaps and larger mismatches over many short gaps and mismatched regions.

The feasibility of cutting protein sequences into fragments, constructing fragment databases, assigning the structures to these fragments and assembling the substructures in order to predict the structure of proteins has been demonstrated in the literature with various methods and fragment sizes [see e.g. Rohl *et al*. (2004), Ruczinski *et al*. (2002), Skolnick *et al*. (2000, 2003), Zhang and Skolnick (2004, 2005)]. Each method uses different selection criteria for the creation of the database, different fragment size distribution and different scoring functions. For example, the Rosetta method (Rohl *et al*., 2004) uses fixed-size fragments of lengths 3 and 9 and is scoring-function independent. The Rosetta database is very large, since its intention is not to cluster similar substructures, but to create a broad distribution of as many 3D sub-structures as possible. Our database, on the other hand, is smaller since it is made of clusters of longer fragments. Our fragments are variable sized and are at least 15 amino acids long. The fragment selection is based on a scoring function that selects only fragments that comprise local minima.

Despite the difference between our BB library and other fragment databases, previous works by different groups have shown the immense potential of a fragment-based cutting and assembly approach in the modeling of protein structures.

PREVIOUS WORK: PAIRWISE SUPERPOSITION ALGORITHMS

Many pairwise superposition methods have been developed. Several alignment methods utilize the DP paradigm. Some of these algorithms use a Double DP algorithm (Cohen, 1997; Taylor, 1999; Taylor and Orengo, 1989). Other methods use a one level DP for an optimal mapping of the residues (Gerstein and Levitt, 1996), for finding matching substructures (Sali and Blundell, 1990) or for a flexible structure alignment (Ye and Godzik, 2003). A series of algorithms focuses on backbone fragment similarities (Ishida *et al*., 2003; Lee *et al*., 2004, 2005; Pei and Grishin, 2004). They first define fragments which can be fixed-length continuous segments from the protein sequence or can be the secondary structure elements. Similar fragment pairs are identified and extended or clustered into more global matches. Shatsky *et al*. (Shatsky *et al*., 2000) proposed a graph theoretic-based technique that uses Dijkstra's shortest path algorithm for a fragment-based alignment. The algorithm aligns two proteins and detects possible hinges with no prior knowledge of the hinge location.

The Geometric Hashing method was introduced by Wolfson *et al*. (Lamdan and Wolfson, 1988; Wolfson, 1990) and later adapted to biological applications by Wolfson and Nussinov (Nussinov and Wolfson, 1991). It enables detection of non-sequential motifs in proteins. The geometric hash table stores redundant transformation-invariant information representing the object and allows fast access to relevant data.

METHODS

The building block clustering

In the clustering procedure we iteratively compare every BB with the representatives of existing clusters using the SSGS algorithm described below. If we find a match between a candidate BB and a cluster representative, we assign the BB to the cluster. If none of the representatives of the clusters match the BB, we create a new cluster with this BB as its representative. The cluster representative is the first BB that opened that cluster. While this clustering method may save computational time it may be inaccurate as BBs in the same cluster can only match the representative, but not one another. To avoid unnecessary comparisons we used two rules: (1) Size difference, the difference between the two compared BB sizes should not be >20% of the size of the smaller of the two BBs. (2) Similar topology, for every BB, we count the number of transitions from a certain secondary structure element to another. A transition is defined as a change in the secondary structure class while moving along the sequence of the BB from the *N*-terminal end. There are two types of secondary structure classes, an α-class that contains only residues that have been assigned as parts of an α-helix and a β-class that contains only residues that have been assigned as parts of a β-sheet. In both classes, we disregarded residues that were assigned to be loops. Therefore, there are two possible transitions: $\alpha \rightarrow \beta$ and $\beta \rightarrow \alpha$. According to the transition type and number of transitions, the BBs are classified into three categories: (1) all α, (2) all β and (3) a mixture of α and β. In the third category, the number of transitions is used to differentiate between the topologies. Specifically, a BB with n transitions is compared only with BBs that have $n \pm 1$ transitions. The topology is assigned using the DSSP algorithm (Kabsch and Sander, 1983).

Secondary structure guided superposition tool

The motivation for this algorithm is to perform a pairwise superposition that is permissive and allows a loose matching between the compared fragments in assigned loop regions (Fig. 1). We can roughly divide the SSGS algorithm into two stages (1) creating a set of transformations of a candidate protein that match fully or partially a target protein and (2) using DP to rank that transformation set. We use the DP score of the superimposition iteratively to refine the superimposition until we reach convergence. Figure 2 depicts an outline of the SSGS algorithm.

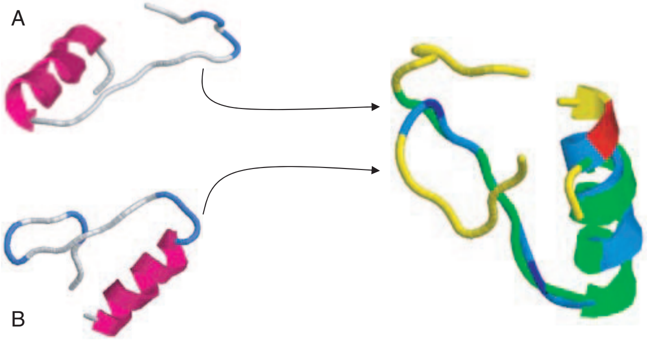


Fig. 1. Given two BB fragments (marked A and B) SSGS superimposes them while giving a greater weight to superimposition of the β-sheet and α-helix assigned residues in an affine mismatch and affine gap manner. The yellow and red colored areas of the proteins show areas that were not matched. In the yellow colored regions, the backbones of the proteins A and B are spatially separated. The red colored area is a gap region, in which the residues of protein A were disregarded. The two proteins are (1) Fructose-1,6-bisphosphatase from Pig, chain B (PDB 1fpg), residues 245–275 and (2) Stromelysin-1 (MMP-3) from Human (PDB 2srt), residues 114–145.

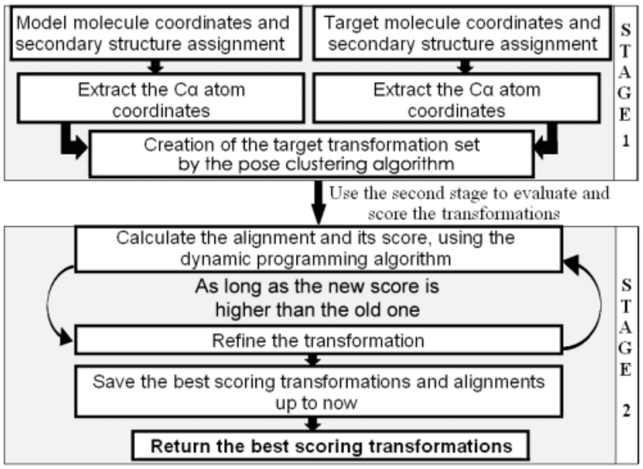

Model molecule coordinates and secondary structure assignment
Target molecule coordinates and secondary structure assignment
Extract the Cα atom coordinates
Extract the Cα atom coordinates
Creation of the target transformation set by the pose clustering algorithm
STAGE 1
Use the second stage to evaluate and score the transformations
Calculate the alignment and its score, using the dynamic programming algorithm
As long as the new score is higher than the old one
Refine the transformation
Save the best scoring transformations and alignments up to now
Return the best scoring transformations
STAGE 2

Fig. 2. A flowchart of the SSGS algorithm.

Stage 1—transformation set creation

(a) Preprocessing and recognition The input to this stage consists of the coordinates of the Cα atoms of two proteins, A and B. Protein A is the target and protein B the model, with lengths of M and N, respectively. The goal of this stage is to generate rigid transformations that match the target fully or partially to the model. To create a transformation set we use the pose clustering method (Ballard, 1981). Pose clustering performs object recognition by determining hypothetical object poses and finds clusters of the poses in the space of the object positions. The poses are the transformations needed to match each triplet of interest points from the target object with each triplet of interest points from the model. The space complexity of pose clustering is lower than the space complexity in the geometric hashing method described above. The space economization causes a higher time complexity. However, because the average object size is small ($\sim$40 interest points) the time complexity difference is small. Pose clustering is based on the notion that a target object that appears in a scene will yield a large cluster of poses close to the correct position of the model object in the scene.

The transformation set creation stage of the algorithm consists of two stages—preprocessing and recognition. Rather than testing all possible transformations, the preprocessing stage matches target triplets only with model triplets that have matching features. The interest points are the Cα atoms. For every non-collinear triplet of the model's Cα atoms (i,j,k), we compute the lengths of the edges of the triangle (i,j,k) and use the lengths as a 3D key to place the indices (i,j,k) in a 3D hash table. In the recognition stage, we iteratively choose non-collinear triplets of the target's Cα atoms (A,B,C) and compute the corresponding edge lengths. We use the lengths as a query key to access the hash table to find possible instances of the model. The query returns all of the model's hashed triangles whose distance from the query key is bounded by a given resolution factor f, where the underlying metric is defined as follows: Let Tr $=$ (i,j,k) be a triangle in the hash table, and let T $=$ (A,B,C) be the query triangle. Denote the lengths of the edges of Tr by ($u1$,$u2$,$u3$), where $u1 = |i \rightarrow j|$, $u2 = |j \rightarrow k|$ and $u3 = |k \rightarrow i|$ and the lengths of the edges of T, similarly by (v1,v2,v3). Then d(Tr,T) $=$ $\max_{1 \leq i \leq 3} |u_i - v_i|$. For every target triplet T $=$ (A,B,C) we compute the transformation needed to transform the triplet to each of the query returned model triplets, thus creating a redundant transformation set. These transformations place a target and model triplet in the same plane that coincides their baricenters. The space complexity of the pose clustering method is $O(M^3)$, which is the number of possible triplets. The time complexity is $O(M^3)$ for the preprocessing stage, plus $O(N^3 * m)$ for the recognition stage. In large proteins, this may result in long computation times because of a possibly extremely large number of transformations. To reduce the size of the transformation set without losing the best transformation, we utilize some heuristics. See Supplementary Material for more details.

(b) Clustering of the transformations We cluster the transformation set by the RMSD to create a non-redundant set. This stage is computationally costly. In a Naïve order dependent clustering algorithm, assuming that the number of clusters and the number of transformations is of the same magnitude, the average time complexity would be O(number of clusters)^2. To refrain from unnecessary comparisons during the clustering and to set a linear time complexity, we use a geometric hash table during the search for a matching cluster for the candidate transformation. The geometric hash table enables us to compare the candidate transformation only against cluster representatives that are potential matches for the candidate transformation. For a detailed description of the transformation clustering method, see Supplementary Material.

Stage 2—alignment creation and evaluation

Each transformation that was created in the previous stage is used as an input to the next stage, where it is used to create an alignment. The alignment and its evaluation are performed by a DP recursion aimed at achieving an affine mismatch topology dependent alignment. This alignment favors a smaller number of long mismatched regions over many short mismatched regions and punishes more strongly for mismatches in α-helix and in β-sheet structures over mismatches in assigned loops. The DP is aimed to achieve the highest score for a given transformation.

Creation of the spatial neighboring relations matrix

Prior to the DP recursion, we translate the input into a spatial neighboring relations matrix (SNRM), $R(M \times N)$. The input is the Cα atomic coordinates of the model and target proteins, with lengths of M and N, respectively. We index the Cα atoms according to their location on the backbone, starting from the N terminal side, and set all the values in R to a mismatch. For every Cα_j ($0 \leq j \leq N$) atom of the model, we compute its Euclidian distance to every Cα_i ($0 \leq i \leq M$) atom of the target. Only if the distance is smaller than radius r and $i - j \leq$ Maximal shift value, we set $R(i,j)$ to a match between Cα_i and Cα_j. The latter condition excludes matches between residues whose indices along the protein sequence are distant from each other. We discard a transformation if the number of matched cells is lower than the maximal shift value, because these transformations cannot yield a satisfying

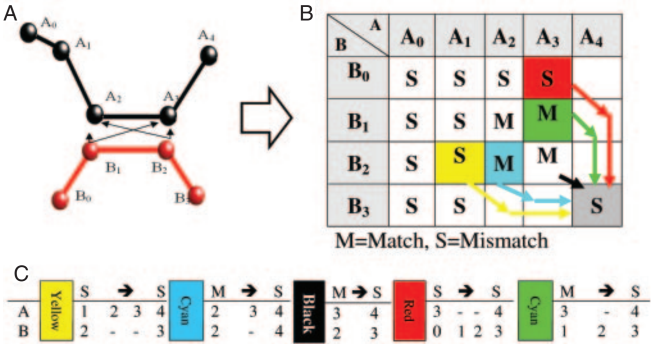


Fig. 3. (**A**) A 2D representation of a transformation of molecules A and B. The black arrows point to the neighboring atoms of atoms B_1 and B_2 from molecule A. (**B** and **C**) A schematic spatial relation matrix of molecules A and B. Each of the five arrows represents a different alignment that starts from different cells and ends in cell (A_4,B_3). The red arrow represents a gap of type mismatch (i.e. starts from a mismatch cell) going from cell (A_3,B_1) to (A_4,B_3). This illustrates an alignment in which we disregard the atoms B_1 and B_2 and align cells (A_3,B_0) and (A_4,B_3) one after the other in the alignment. Because cell (A_3,B_1) is a mismatch and cell (A_4,B_3) is a mismatch then such an alignment depicts a continuation of a mismatch region. The green arrow represents a gap of type match (i.e. starts from a match cell) going from cell (A_3,B_1) to cell (A_4,B_3). This gap aligns cells (A_3,B_1) and (A_4,B_3) one after another, thus opening a mismatch region. The black arrow represents the diagonal alignment going from cell (A_3,B_2) to cell (A_4,B_3). Aligning these cells one after the other represents opening a mismatch region. The cyan and yellow arrows represent a match and mismatch gaps respectively from the leftward side. (C) The alignments that each of the arrows depicts: the red arrow alignment represent continuing a mismatch region (S→S), the green arrow alignment represents opening a new mismatch (M→S).

alignment with enough matches. We set the maximal shift value to a third of the longest protein length, and the radius r to 3 Å.

Alignment creation

To simplify the description, we first introduce a secondary structure independent affine mismatch and gap alignment and later describe how the secondary structure dependency is added into the algorithm.

We compare our scoring system with Gotoh's affine gap penalty algorithm (Gotoh, 1982). Gotoh's algorithm calculates three values at each recursion stage. Two of the values represent the score of alignments ending with an open gap that goes through the current cell. The third value represents the best alignment that ends in the current cell and is dependent on the maximal value of the previously calculated three values of its upper left diagonal cell. At every stage it checks whether opening a new gap yields a higher alignment score than continuing the already opened gap alignment and whether matching the compared indexed objects will yield a higher score than the former values.

Our affine gap and mismatch score does not only depend on the location of the cell the gap starts from and the size of the gap (as in Gotoh's affine gap penalty algorithm), but also on whether the cell the gap starts from is a mismatch or a match. For example, in Figure 3, given a matrix $D_{(M+1,N+1)}$, if cell $D_{(i,j)}$ is considered as a mismatch cell, then cell $D_{(i,j)}$ can be either the beginning of a new mismatch region or a continuation of an existing mismatched region (view the red and green arrows alignments in Fig. 3). If the cell that leads to $D_{(i,j)}$ is considered as a match, aligning these cells one after the other will open a new mismatched region. For example, aligning the diagonal cell $D_{(i-1,j-1)}$ and $D_{(i,j)}$ (view the black arrow alignment in Fig. 3) will open a new mismatched region. In some cases a better alignment score can be achieved for closing a gap of type 'mismatch' that starts at cell $D_{(i-3,j-1)}$ (thus closing a mismatch region), over closing a gap of type 'match' that starts from cell $D_{(i-2,j-1)}$ (thus continuing a match region). As illustrated in these examples, a path that starts from a mismatch and a path that starts from a match may have different impact on the alignment score, hence at every cell $D_{(i,j)}$, we consider both possibilities and differentiate between two types of gaps: gaps that start from a mismatch cell and gaps that start from a match cell. As seen in the figure, the value that represents the best alignment that ends in cell $D_{(i,j)}$ is the maximum among five values: the leftward and upward cells each contributing two values gap (where each gap starts from a different cell) and the diagonal alignment value. Hence, when we update the relevant gaps for every $D_{(i,j)}$ we first identify the new gap type, [i.e whether the cell ($D_{(i-1,j-1)}$) the gap starts from is a 'mismatch' or a 'match' cell] and whether opening the new gap yields a better alignment score over the existing gap.

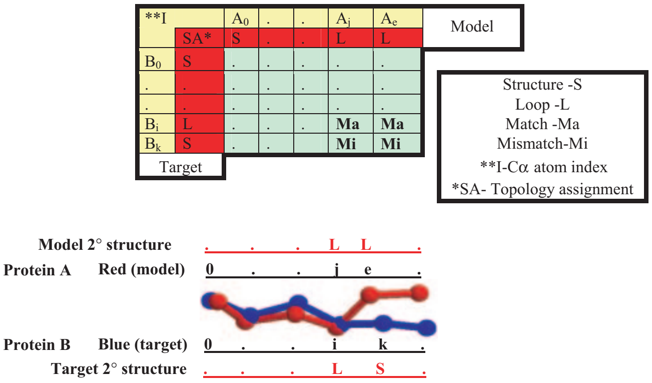


Fig. 4. The table illustrates the criteria that influence the topology factor for two proteins A and B: the 2° structure assignment (red cells), and the spatial relation matrix (green cells). For example, cell B_i, A_j is an alignment of two loop assigned residues and is regarded as a match due to the spatial neighboring of Cα atoms B_i and A_j according to the spatial relation matrix. The figure at the bottom illustrates a graphic representation of the table. It shows the indices (black), the 2° structure assignment (red), the match between A_j and B_i Cα atoms and the mismatch between the A_e and B_k Cα atoms.

The secondary structure dependency

The secondary structure is introduced by further differentiating between cell types. Residues that belong to β-sheets or α-helices are considered as a 'structure'. Otherwise they are considered as a 'loop'. We assess the assignment by DSSP (Kabsch and Sander, 1983). These definitions of the assignment give rise to three mismatch cell types (loop to loop, loop to structure and structure to structure mismatch cells) and similarly three types of match cells. For example cell (A_j, B_i) in Figure 4 exemplifies a matched cell of type loop to loop.

The secondary structure dependency of the alignment is manifested in the topology factor, which is the payoff given by aligning two cells one after another depending on (1) the secondary structure assignment of the aligned residues in the cell, (2) whether the cell is a match or a mismatch and (3) the route progress: either opening a mismatched region, continuing a matched region, continuing a mismatched region or closing a mismatched region. As seen in the example in Figure 4, the final payoff for aligning the cells (B_i,A_j) and (B_k,A_e) is calculated considering the assignment and the spatial relations of the aligned residues. Cell (B_i,A_j) aligns two loop-assigned Cα atoms that are spatial neighbors. Cell (B_k,A_e) aligns a loop and structure-assigned Cα atoms that are not spatial neighbors. Since cell (B_i,A_j) is a match, and cell (B_k,A_e) is a mismatch, aligning the two cells opens a mismatched region in the alignment. The final payoff for aligning cells (B_i,A_j) and (B_k,A_e) is the average of the payoffs for opening a mismatch in a loop-to-loop residues alignment and a structure-to-loop residues alignment.

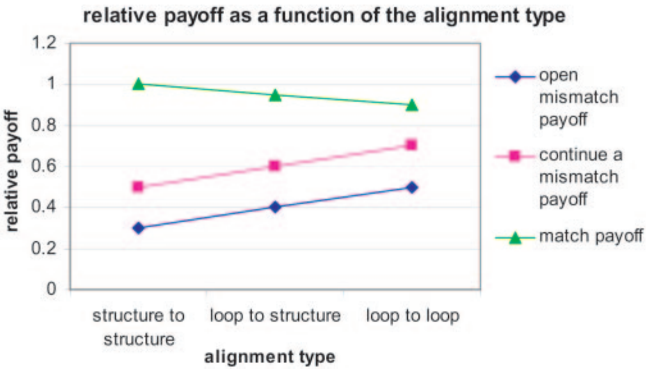


Fig. 5. The graph illustrates the relative payoffs given in the alignment stage of the SSGS algorithm as a function of the alignment type.

Payoff factors

The payoffs are the weights that rank the alignment. The ratio between the payoffs endows upon the alignment its affine mismatch secondary structure-dependent nature. Figure 5 illustrates the payoff combination optimized for a test set of 200 proteins. The ratio payoff has the following criteria:

(1) The payoff for opening a mismatch in a loop-to-loop is higher than that in a structure-to-structure alignment. Similarly, the payoff for continuing a mismatch in a loop-to-loop is higher than that in a structure-to-structure alignment. These ratios guarantee that the best alignment will favor mismatches in loops over mismatches in α-helix and β-sheets.

(2) The payoff for a match in a structure-to-structure is higher than a match in a loop-to-loop alignment. This reflects the tendency for matches in structure-assigned residues.

(3) The payoff for opening a mismatch is always smaller than continuing an already opened mismatch. This favors few long mismatch regions over many short ones.

(4) We expect loop-to-structure alignment to appear more often in the margins of the assigned secondary structure, where the assignment might be imprecise. To ensure some flexibility in the alignment, the payoffs of loop-to-structure are the average of the loop-to-loop and structure-to-structure alignment payoffs.

Implementation

We create three matrices: $\mathrm{Opt}_{(M+1,N+1)}$, $\mathrm{Left}_{(M+1,N+1,6)}$ and $\mathrm{Up}_{(M+1,N+1,6)}$. The Opt matrix holds the value for an optimal alignment that ends in the current cell. The matrices Up and Left are composed of six values, each for a different cell type for the optimal path that ends with a gap leading to the current cell, depending on the topology of the aligned residues and the match or mismatch.

We modify Gotoh's affine gap penalty to achieve an alignment satisfying both secondary-structure-dependent affine mismatch and affine gap characteristics. At every stage of the DP we compute 13 values, as the outcome of our differentiation between six cell types (determined by the match/mismatch state and the secondary structure alignment) at which a gap region can start from (6 values are calculated for each of the Left and Up matrices and the 13th value is the optimal score). At every stage of the recursion, we save the highest payoffs for these gap types (the gap type is determined by the cell it starts from). As the recursion progresses, we check the current cell type and determine whether opening a gap at the current cell type achieves a higher payoff.

Trace back: Given matrices Opt, Left and Up, find the type of the highest value among the cells Opt (M,N), Left(M,N) and Up(M,N). According to the matrix k, which contains the highest value, we set the next cell in the alignment. For a detailed description of the algorithm implementation, see Supplementary Material.

1bhs	Red	-R-TV-VLITGCSSGIGLHLAVRLASD--PSQSF----KV-Y---AT
1gga	Green	--TI-KVGING-FGRIGRMVFQALCDD--G--LLGNEIDVVA---V-
1gyp	Yellow	-P-I-KVGING-FGRIGRMVFQAICDQ--G--LIGTEIDVVA---V-
1dnp	Orange	-H-LF-YNYQ--YEVNERARDVEVERAL--RNVV---CEG-FDDS--
2cmd	Blue	---KV-AVL-GAAGGIGQALALLLKTQLP-SGSE----LS-L---YD
Conservation		13115131339113366333331311311553301115131111131

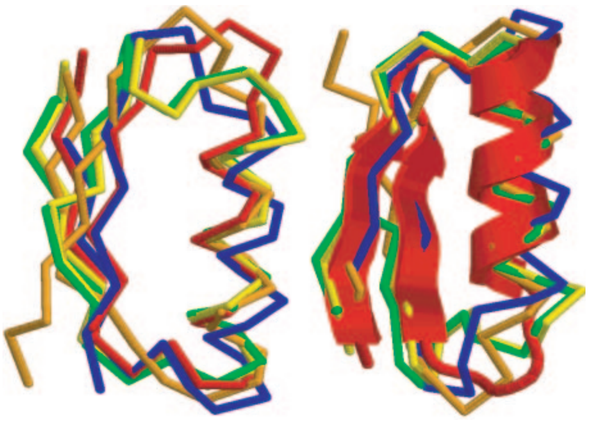


Fig. 6. A multiple structural alignment of a BB cluster from two views. The representative of this cluster is the BB consisting of residues 2–35 of the human estrogenic 17 beta-hydroxysteroid dehydrogenase (PDB: 1bhs, red chain) which is considered to be the average structure for the cluster. The sequence alignment is shown above. The building blocks are 1bhs, (residues 2–35) Human Estrogenic 17 β-Hydroxysteroid Dehydrogenase; 1dnp, (residues 38-131) Deoxyribodipyrimidine Photolyase; 1gga, (residues 1–35) Glycosomal glyceraldehyde-3-phosphate dehydrogenase; 1gyp, (residues 2–36) Glycosomal glyceraldehyde-3-phosphate dehydrogenase, 2cmd, (residues 2–34) Malate Dehydrogenase

RESULTS AND DISCUSSION

Clustering of the building blocks

We clustered 67 971 BBs from 8617 protein chains into 14 401 clusters with an average of 4.7 BBs per cluster. The choice of the protein chains was based on a non redundant representation of the PDB (Fischer *et al*., 1995). Of the clusters 62% (8933 clusters) had more than one member. As expected, the ratio between the sizes of the BB clusters and the BB lengths exhibited an exponential growth in the size of the BB cluster as the length of the BB representative shortens. In most of the clusters the majority of the members belonged to the same SCOP family, implying a high sequence resemblance. Both high sequence and high structural similarity are crucial for the template's ability to reflect a sequence-structure pattern that is statistically identifiable in a sequence-profile search. On the one hand, the sequence similarity has to be high. On the other hand, it has to reflect some of the sequence polymorphism the BB cluster structure exhibits.

Figure 6 shows the multiple structural alignments and the multiple sequence alignment (MSA) of a cluster whose representative is human estrogenic 17 beta-hydroxysteroid dehydrogenase (residues 2–35). The multiple structural alignment of the cluster exhibits a low RMSD (2.0 Å), implying that there is a high structural similarity. However, the MSA exhibits a low sequence conservation [calculated by Al_2CO (Pei and Grishin, 2001)]. This example illustrates an already known phenomenon, according to which the structure is often better conserved than the sequence. Further, the structural flexibility at the BB ends and at the inter-BB area may be important for enabling the combinatorial trial-association process of the BBs as the protein folds. As expected, the least structurally conserved regions are the BB ends, which are often loop regions.

The BB folding model leads to an important question: Can the BBs be referred to as stand-alone units, independent of their structural environment? Will a BB show similar properties if it is taken out of its structural context and placed elsewhere? To address this question we use the clusters, which contain an abundance of data. The most interesting clusters are the ones containing many BBs from different protein families and which vary greatly in their sequences. We encountered many such clusters (Haspel *et al*., 2003a; Wainreb, 2005), illustrating that the BBs can usually be viewed as independent units, regardless of their environmental context. If this is true, the BB clusters can be a powerful tool that helps to reduce the complexity of the protein structure prediction via the hierarchical three-stage protein folding scheme described above.

The SSGS algorithm

The BB clustering and the SSGS algorithm used for BB alignment during the clustering were implemented on an AMD 1700 MHz based workstation. We ran the SSGS algorithm over a large number of examples (approximately billion comparisons). The average running time for finding the optimum geometric alignment between two protein chains with a length of 40 amino acids, the average length of the BB in the BB database, was 3.5 s.

Figure 7 shows three examples of the results of the SSGS algorithm, when applied to BBs. All three cases show <20% sequence identity. In Figure 7B and C the two BBs are derived from proteins of different super-families. In Figure 7A the two BBs are from proteins of the same family, but from different regions of the protein. This shows that the SSGS algorithm finds good structural matches even in cases where the sequence homology is weak. As seen in the figures, the secondary structure match is preferred over matches of loops and unassigned areas even if as a result, the overall RMSD of the match is higher. Since loop regions are less conserved than secondary structure assigned regions, a pairwise matching algorithm that prefers secondary structure matching to loop matching is more likely to discover matches that are significant to the conservation and retention of the protein structure.

The alignment created by SSGS is a balance between four alignment goals: (1) to maximize the number of pairs of equivalent main chain atoms, (2) to minimize the alignment error i.e. the root mean square deviation (RMSD) between the aligned atom pairs, (3) to align the two fragments while giving a higher preference to aligning the secondary structures assigned residues, and (4) to favor few long mismatched regions over many short mismatched regions.

Comparison with other algorithms

In order to test the SSGS performance relative to other algorithms, we compared its results with those of three other structural alignment tools—MASS (Dror *et al*., 2003), MutiProt (Shatsky *et al*., 2002) and DALI (Holm and Sander, 1993). These alignment tools represent two methodologies of structurally based comparison. MultiProt and DALI, on the one hand, align the proteins, disregarding their secondary structure and their sequence order. MASS, on the other hand, aligns the protein structures almost solely based on their secondary structures. Compared with these methods SSGS exhibits a compromise genre that gives a more flexible attribute to secondary structure and non-secondary structure assigned Cα atoms.

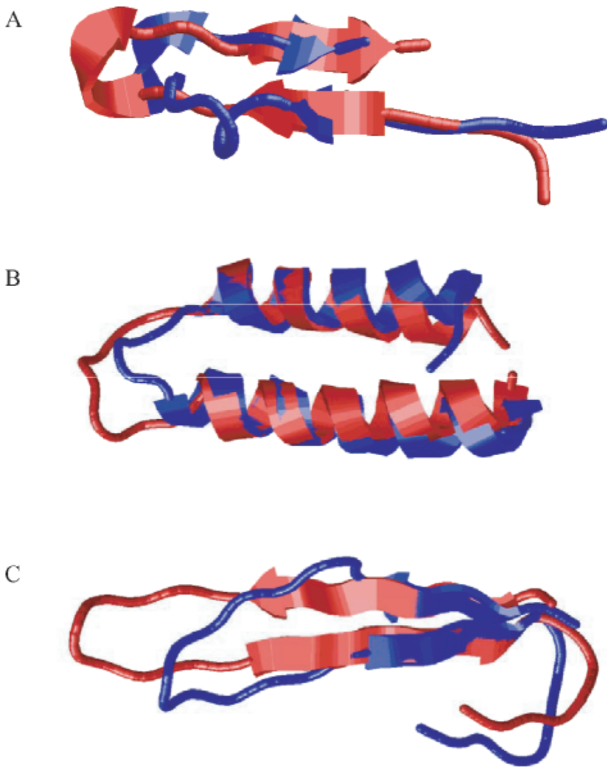


Fig. 7. Examples of the results of the SSGS algorithm when applied to BBs. In each case the model BB is colored in blue and the target is colored in red. The BBs are (**A**) Model: Aspartate aminotransferase from *Escherichia coli* (PDB: 1asl), chain A, residues 249–269. Target: Aspartate aminotransferase from chicken (PDB: 2cst), chain B, residues 63–82. (**B**) Model: Phycocyanin beta subunit from cyanobacterium (PDB: 1cpc), chain L, residues 48–96. Target: Cyclin H (mcs2) from human (PDB: 1jkw), residues 126–174. (**C**) Model: Transglutaminase from human (PDB: 1ggt), chain A, residues 2–34. Target: immunoglobulin from mouse (PDB: 1clo), chain L, 516–549.

We performed a large-scale iterative pairwise alignment of a dataset of protein segments taken from our BB library using SSGS, MASS, MultiProt and DALI. The subset of protein segments was selected trying to maintain two database criteria: uniform distributions of segment sizes and of secondary structure percentage (i.e. the fraction of the number of residues assigned as α-helix or β-sheet out of the total number of residues). Overall, we selected a representative set made of 263 pairs of protein segments from the original BB database. To evaluate the structural alignments we used STACCATO, which is a novel MSA tool (Shatsky *et al*., 2006). Given a structural alignment, it incorporates both the sequence and the structural information into its resultant MSA.

The segment pairs in the dataset were each aligned by MASS, MultiProt, DALI and SSGS and compared according to STACCATO's identity score. We denoted the identity score for the alignment of protein segments A and B $\mathrm{Id}_{\mathrm{SSGS}}^{A,B}, \mathrm{Id}_{\mathrm{MASS}}^{A,B}, \mathrm{Id}_{\mathrm{MULTI}}^{A,B}, \mathrm{Id}_{\mathrm{DALI}}^{A,B}$

Table 1. The average differences in performance between SSGS and three other structural alignment methods—MASS, MultiProt and DALI

Method	Identity score difference with SSGS
MASS	0.61 ± 4.01
MultiProt	0.53 ± 3.71
DALI	0.79 ± 6.4

The difference in the performance is expressed as the difference between the identity scores of each comparison, obtained by STACCATO (see Results section). The data was generated running the three algorithms on 400 protein segment pairs. Histogram representations of the results can be found in Figure 8.

for SSGS, MASS, MultiProt and DALI, respectively. We evaluated the results of SSGS comparing to the other methods according to the difference in the identity scores achieved by the three methods for each protein pair, i.e.: $\mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{SSGS-MASS}} = \mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{SSGS}} - \mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{MASS}}$, $\mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{SSGS-MULTI}} = \mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{SSGS}} - \mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{MULTI}}$ and $\mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{SSGS-DALI}} = \mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{SSGS}} - \mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{DALI}}$.

In ~80% of the comparisons, there were no substantial differences between the results obtained by the different methods. As expected, we encountered more differences between the methods as the average identity value $[(\mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{SSGS}} + \mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{MASS}} + \mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{MULTI}} + \mathrm{Id}^{\mathrm{A,\,B}}_{\mathrm{DALI}})/4]$ of the alignments decreased. This observation emphasizes the fact that as the compared proteins become more evolutionarily distant, it is more important to choose the correct alignment tool for the problem. As can be seen in Table 1, on average, SSGS performed better than MASS, MultiProt and DALI but this result is not statistically significant. However, in the cases in which SSGS was favorable to MultiProt, the average difference in the identity score ($\mathrm{Id}_{\mathrm{SSGS-MULTI}}$) was 5.5% compared with only 1.6% in the cases in which SSGS was inferior to MultiProt (i.e. $\mathrm{Id}_{\mathrm{SSGS-MULTI}} < 0$). This pattern can be also seen with regard to MASS. In the cases in which SSGS was favorable to MASS ($\mathrm{Id}_{\mathrm{SSGS-MASS}} > 0$) the average difference in the alignment identity was 8.3%. In the cases in which SSGS was inferior to MASS (i.e. $\mathrm{Id}_{\mathrm{SSGS-MASS}} < 0$) the average identity difference was 1.7%. In the cases in which SSGS was favorable to DALI ($\mathrm{Id}^{A,B}_{\mathrm{SSGS-DALI}} > 0$) the average difference in the alignment identity was 15%. In the cases in which SSGS was inferior to DALI (i.e. $\mathrm{Id}^{A,B}_{\mathrm{SSGS-DALI}} < 0$) the average identity difference was 2%. A graphic representation of these results can also be seen in Figure 8 where the comparison of SSGS with the other three algorithms is shown in the form of a histogram.

In order to characterize those cases in which it would be preferable to use SSGS, we studied how the difference in the number of secondary structures of the aligned proteins affected the performance of the four methods. Cases in which SSGS results were favorable compared with MASS, MultiProt or DALI, the difference between the number of secondary structure elements of the aligned protein was significantly smaller. Unlike MASS, MultiProt or DALI that are sequence order independent, SSGS is a sequence order dependent algorithm. This feature guides the alignment results of SSGS to the detection of order-dependent motifs which are better conserved among topologically related protein segments. Evolutionarily, the loop regions are poorly conserved and exhibit a high rate of insertions and deletions. Thus, finding consensus residues might require a better alignment of the secondary structures that are more conserved, over aligning the loop assigned regions.

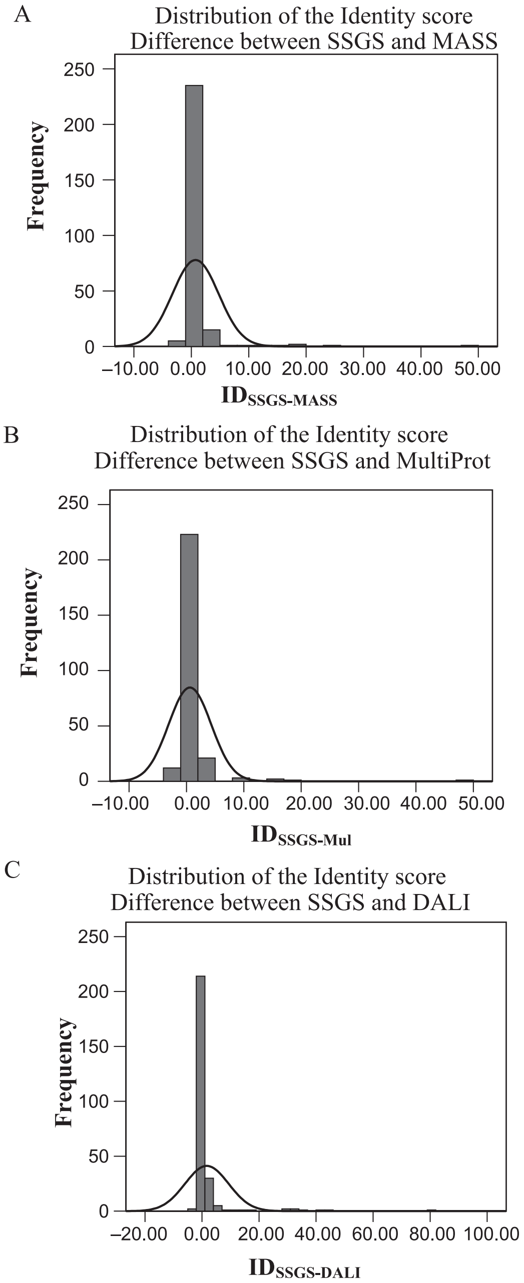
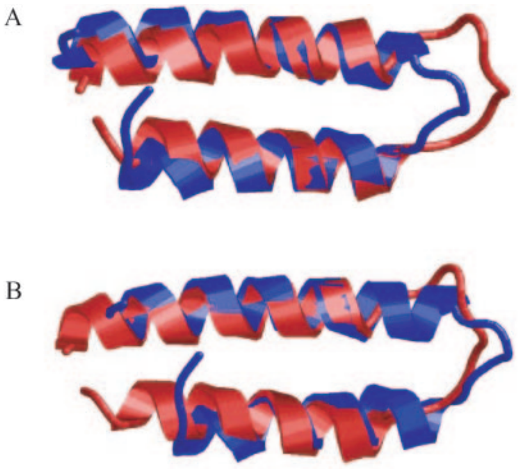


Fig. 8. A histogram representation of the comparisons between SSGS and three other structural alignment methods—MASS (**A**), MultiProt (**B**) and DALI (**C**). This is a histogram presentation of the results displayed in Table 1. The difference in the performance is expressed as the difference between the identity scores of each comparison, obtained by STACCATO (see Results section). The data was generated running the three algorithms on 400 protein segment pairs.

This makes a secondary structure guided alignment more significant when attempting to find similar structural patterns among proteins from different families, as is the case of the clustering of the BBs. An example of this preference can be seen in Figure 9, where we can view two different alignments of the same two chains created by (1) SSGS and (2) Multiprot. The two protein chains are BB fragments



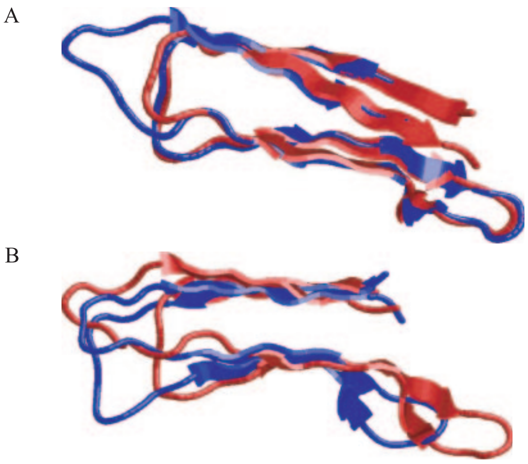
```
SSGS - Identity: 7% 4 NormScore: 3.76761
  Alignment:
 jkw    48> LEPHEEMTLCKYYEKRLLEFCSV---FKPAMPRSVVGTACMYFKRFYLNNSV
 cpcL  126> --GSSVAVGVQKMKDAALAIAGDTNGITRGDCASLMAEVASYFDKAAAAVA-
                             *               *       **

MultiProt - Identity: 1% 1 NormScore: 3.99962
  Alignment:
jkw    48> --LEPHEEMTLCKYYEKRLLEFCSVFKPAMPRSVVGTACMYFKRFYLNNSAAVA
cpcL  126> GSSVAVGVQKMKDAALAIAGD-TNGIT----RGDCASLMAEVASYFDKAAV---
                                          *
```

Fig. 9. An example of the matching of two BBs using two different pairwise matching algorithms. (**A**) Alignment using the SSGS algorithm. (**B**) Alignment using the MultiProt algorithm. The sequence alignment of each match is shown below. The two BBs are (1) Cyclin H (mcs2) from human (PDB: 1jkw), residues 126–174 (blue). (2) Phycocyanin beta subunit from cyanobacterium (PDB: 1cpc), chain L, residues 48–96 (red).

that were cut from human Cyclin H (mcs2), residues 126–174 and Phycocyanin beta subunit from cyanobacterium, chain L, residues 48–96. Both protein segments contain a helix–loop–helix assigned region. As the figure shows, the SSGS resulting superposition exhibits an alignment in which the matching of the helix assigned residues was preferred over an alignment with more residues and a lower overall RMSD which does not fully align the helix assigned residues, as observed in the resulting superposition created by MultiProt. As shown in Figure 9, the sequence alignment of the match obtained by the SSGS algorithm is better than the sequence alignment of the match obtained by MultiProt for the same model and target. This stems from the fact that the SSGS aligned the secondary structure elements better than MultiProt, at the expense of a poorer loop alignment.

Figure 10 provides the alignment of two BBs using SSGS and MASS. The aligned segments are BB fragments cut from the human tumor necrosis factor (TNF) (PDB: 1tnf), residues 60–121 and the extra-cellular domain of the human CD40 ligand (PDB: 1aly), residues 55–114. Both protein segments contain two loop regions and two secondary structure regions. As can be seen, the alignment obtained by SSGS is much better both structurally and by means of identity score, than the alignment obtained by MASS. This stems from the fact that MASS first aligns secondary structure elements and later attempts to find the best alignment within this framework, while SSGS aligns all the residues in one stage, which may result in an alignment that prefers loop regions over secondary structure regions to maximize the overall score. In addition, a secondary-structure-based algorithm such as MASS is likely to fail in finding a match in cases where a BB contains few or no secondary structure elements whereas SSGS, despite preferring the alignment of secondary structure elements, can still proceed. The structure guided sequence alignment shown in Figure 10 shows that the SSGS guided alignment prefers one larger gap in the loop region over two smaller gaps as in the MASS case. To sum, our tests indicate that SSGS is a good compromise between the two approaches represented by secondary–structure-dependent methods such as MASS and residue-based sequence order independent matching methods such as MultiProt or DALI.



```
SSGS  - Identity :16% 11 NormScore :3.1967
Alignment:
cdaA  60> ---SQVLFKGQGCPSTHVLLTHTISRIAVSYQTKVNLLSAIKSPCQRETPEGAEAKPWYEPIYLG---
tnfA  55> YVYTQVTFCSNREPSSQRPFIVGLWLKP-SIGSERILLKAANTHSSSQ-------LCEQQSVHLGGVF
              ** *     **              *      ** *                       **

MASS  - Identity :7% 5 NormScore :4.25044
Alignment:
cdaA   55> YVYTQVTFCSNREPSSQRPFIVGLWLKPSI----GSERIL--LKAANTHSSSQLCE-QQSVHLGGVF
tnfA   60> SQVLFKGQGC---PS--THVLLTHTISRIAVSYQTKVNLLSAIKSPCQRETPEGAEAKPWYEPIYLG
                        **                        *   *           *
```

Fig. 10. An example of the matching of two BBs using two different pairwise matching algorithms. (**A**) Alignment using the SSGS algorithm. (**B**) Alignment using the MASS algorithm. The sequence alignment of each match is shown below. The two BBs are (1) TNF from human (PDB: 1tnf), chain A, residues 60–121 (blue). (2) The extra-cellular domain of the human CD40 ligand (PDB: 1aly), residues 55–114 (red).

CONCLUSIONS

Here, we present a new method for fragment structure alignment. Our method comprises a novel technique for achieving an affine mismatch alignment. The method successfully performs pairwise alignment, while considering the topology of the aligned chains. The reference to the secondary structure assignment avoids an exhaustive scan of the transformation space without missing the optimal alignment. The SSGS algorithm was used for a large-scale clustering of protein fragments derived from the PDB. Considering the accommodating nature of this method, we propose its use while modifying the parameters according to the evolutionary distance between the aligned proteins. Instead of disregarding the loop-assigned residues or award loop residues the same weight as the structure-assigned residues, it is possible to adjust the weight of the loop-assigned residues in the alignment to reflect the fact that they are less conserved evolutionarily.

In particular, SSGS is able to efficiently and robustly create structurally similar clusters of fragments of protein chains when these may (largely) consist of loops. This ability of SSGS makes it a useful tool towards structure prediction via modeling of local structures followed by their assembly. In cases where there are no sequentially similar chains with available structures which can be used in homology modeling, modeling of fragments may prove a sound strategy. In such an approach, a library rich in clusters of structurally similar BB fragments would be key for large-scale protein structure prediction.

ACKNOWLEDGEMENTS

The authors thank Maxim Shatsky and Yuval Inbar for assistance. The authors also thank Drs C.-J. Tsai, K. Gunasekaran and H.-H. (G.) Tsai for discussions. The computation times are provided by the National Cancer Institute's Frederick Advanced Biomedical Supercomputing Center. The research of R.N. in Israel and H.W. has been supported in part by the 'Center of Excellence in Geometric Computing and its Applications' funded by the Israel Science Foundation (administered by the Israel Academy of Sciences). This project 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. This research was supported [in part] by the Intramural Research Program of the NIH, National Cancer Institute, Center for Cancer Research. 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 US Government. Funding to pay the Open Access publication charges for this article was provided by SAIC, NCI-Frederick.

Conflict of Interest: none declared.

REFERENCES

Baldwin,R.L. (1994) Matching speed and stability. *Nature*, **369**, 183–184.

Baldwin,R.L. (1995) The nature of protein folding pathways: the classical versus the new view. *J. Biomol. NMR*, **5**, 103–109.

Ballard,D.H. (1981) Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit*., **13**, 111–122.

Ben-Naim,A. (1980) Hydropobic Interactions. Plenum Press, New York, pp. 3–15.

Bryngelson,J.D. and Wolynes,P.G. (1989) Intermediates and barrier crossing in a random energy model with applications to protein folding. *J. Phys. Chem*., **93**, 6902–6915.

Bryngelson,J.D. *et al*. (1995) Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins*, **21**, 167–195.

Bryngelson,J.D. and Wolynes,P.G. (1989) Intermediates and barrier crossing in a random energy model with applications to protein folding. *J. Phys. Chem*., **93**, 6902–6915.

Cohen,G.H. (1997) ALIGN: a program to superimpose protein coordinates, accounting for insertions and deletions. *J. Appl. Cryst*., **30**, 1160–1161.

Dill,K.A. and Chan,H.S. (1997) From Levinthal to pathways to funnels. *Nat. Struct. Biol*., **4**, 10–19.

Dill,K.A. (1990) Dominant forces in protein folding. *Biochemistry*, **29**, 7133–7155.

Dill,K.A. *et al*. (1995) Principles of protein folding—a perspective from simple exact models. *Protein Sci*., **4**, 561–602.

Dror,O. *et al*. (2003) Multiple structural alignment by secondary structures: algorithm and applications. *Protein Sci*., **12**, 2492–2507.

Fischer,D. *et al*. (1995) A 3D sequence-independent representation of the protein data bank. *Protein Eng*., **8**, 981–997.

Gerstein,M. and Levitt,M. (1996) Using iterative dynamic programming to obtain pairwise and multiple alignments of protein structures. *Proc. Int. Conf. Intell. Syst. Mol. Biol*., **4**, 59–67.

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol*., **162**, 705–708.

Haspel,N. *et al*. (2003) Hierarchical protein folding pathways: a computational study of protein fragments. *Proteins*, **51**, 203–215.

Haspel,N. *et al*. (2003) Reducing the computational complexity of protein folding via fragment folding and assembly. *Protein Sci*., **12**, 1177–1187.

Holm,L. and Sander,C. (1993) Protein structure comparison by alignment of distance matrices. *J. Mol. Biol*., **233**, 123–138.

Inbar,Y. *et al*. (2005) Prediction of multimolecular assemblies by multiple docking. *J. Mol. Biol*., **349**, 435–447.

Ishida,T. *et al*. (2003) Development of an *ab initio* protein structure prediction system ABLE. *Genome Inform. Ser. Workshop Genome Inform*., **14**, 228–237.

Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.

Karplus,M. *et al*. (1995) Comment: kinetics of protein folding. *Nature*, **373**, 664–665.

Karplus,M. *et al*. (1997) The Levinthal paradox, yesterday and today. *Fold Des*., **2**, S69–S75.

Karplus,M. and Shakhnovitch,E.I. (1992) Protein folding: Theoretical studies of thermodynamics and dynamics. In Creighton,T.E. (ed.), *Protein Folding*. W.H. Freeman and Company, New York, pp. 127–195.

Lamdan,Y. and Wolfson,H.J. (1988) Geometric hashing: a general and efficient modelbased recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision*. Tampa, Florida, USA, pp. 238–249.

Lee,J. *et al*. (2004) Prediction of protein tertiary structure using PROFESY, a novel method based on fragment assembly and conformational space annealing. *Proteins*, **56**, 704–714.

Lee,J. *et al*. (2005) Protein structure prediction based on fragment assembly and parameter optimization. *J. Biophys. Chem*., **115**, 209–214.

Lesk,A.M. and Rose,G.D. (1981) Folding units in globular proteins. *Proc. Natl Acad. Sci. USA*, **78**, 4304–4308.

Levinthal,C. (1968) Are there pathways for protein folding? *J. Chem. Phys*., **65**, 44–45.

Martinez,J.C. *et al*. (1998) Obligatory steps in protein folding and the conformational diversity of the transition state. *Nat. Struct. Biol*., **5**, 721–729.

Nussinov,R. and Wolfson,H.J. (1991) Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *Proc. Natl Acad. Sci. USA*, **88**, 10495–10499.

Onuchic,J.N. *et al*. (1995) Toward an outline of the topography of a realistic protein-folding funnel. *Proc. Natl Acad. Sci. USA*, **92**, 3626–3630.

Onuchic,J.N. *et al*. (1996) Protein folding funnels: the nature of the transition state ensemble. *Fold Des*., **1**, 441–450.

Pace,C.N. *et al*. (1996) Forces contributing to the conformational stability of proteins. *FASEB J*., **10**, 75–83.

Pei,J. and Grishin,N.V. (2001) AL2CO: calculation of positional conservation in a protein sequence alignment. *Bioinformatics*, **17**, 700–712.

Pei,J. and Grishin,N.V. (2004) Combining evolutionary and structural information for local protein structure prediction. *Proteins*, **56**, 782–794.

Rohl,C.A. *et al*. (2004) Protein structure prediction using Rosetta. *Methods Enzymol*., **383**, 66–93.

Ruczinski,I. *et al*. (2002) Distributions of beta sheets in proteins with application to structure prediction. *Proteins*, **48**, 85–97.

Sali,A. and Blundell,T. (1990) Definition of general topological equivalance in protein strucutres: a procedure involving comparison of properties and relationships through similated annealing and dynamic programming. *J. Mol. Biol*., **212**, 403–428.

Shatsky,M. *et al*. (2000) Alignment of flexible protein structures. *Proc. Int. Conf. Intell. Syst. Mol. Biol*., 329–343.

Shatsky,M. *et al*. (2002) MultiProt—a multiple protein structural alignment algorithm. In *Lecture Notes in Computer Science*. Springer Verlag, Heidelberg, Germany, pp. 235–250.

Shatsky,M. *et al*. (2006) Optimization of multiple-sequence alignment based on multiple-structure alignment. *Proteins*, **62**, 209–217.

Skolnick,J. *et al*. (2000) Derivation of protein-specific pair potentials based on weak sequence fragment similarity. *Proteins*, **38**, 3–16.

Skolnick,J. *et al*. (2003) TOUCHSTONE: a unified approach to protein structure prediction. *Proteins*, **53**, 469–479.

Struthers,M.D. *et al*. (1996) Design of a monomeric 23-residue polypeptide with defined tertiary structure. *Science*, **271**, 342–345.

Taylor,W.R. (1999) Protein structure comparison using iterated double dynamic programming. *Protein Sci.*, **8**, 654–665.

Taylor,W.R. and Orengo,C.A. (1989) Protein structure alignment. *J. Mol. Biol.*, **208**, 1–22.

Tsai,C.J. *et al.* (2000) Anatomy of protein structures: visualizing how a one-dimensional protein chain folds into a three-dimensional shape. *Proc. Natl Acad. Sci. USA*, **97**, 12038–12043.

Tsai,C.J. and Nussinov,R. (2001) The building block folding model and the kinetics of protein folding. *Protein Eng.*, **14**, 723–733.

Tsai,H.H. *et al.* (2004) *In silico* protein design by combinatorial assembly of protein building blocks. *Protein Sci.*, **13**, 2753–2765.

Tsong,Y. *et al.* (1972) Properties of the refolding and unfolding reactions of ribonuclease A. *Proc. Natl Acad. Sci. USA*, **69**, 1809–1812.

Wainreb,G. (2005) *Faculty of medicine*, Master's thesis. Tel Aviv University, Tel-Aviv.

Wolfson,H.J. (1990) Modelbased recognition by geometric hashing. In *Proceedings of the First European Conference on Computer Vision.* Antibes, France, pp. 526–536.

Ye,Y. and Godzik,A. (2003) Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, **19** (Suppl. 2), II246–II255.

Yue,K. *et al.* (1995) A test of lattice protein folding algorithms. *Proc. Natl Acad. Sci. USA*, 325–329.

Zhang,Y. and Skolnick,J. (2004) Automated structure prediction of weakly homologous proteins on a genomic scale. *Proc. Natl Acad. Sci. USA*, **101**, 7594–7599.

Zhang,Y. and Skolnick,J. (2005) The protein structure prediction problem could be solved using the current PDB library. *Proc. Natl Acad. Sci. USA*, **102**, 1029–1034.