FAST: A Novel Protein Structure Alignment Algorithm

Jianhua Zhu[1] **and Zhiping Weng**[1,2*]
[1]*Bioinformatics Program, Boston University, Boston, Massachusetts*
[2]*Biomedical Engineering Department, Boston University, Boston, Massachusetts*

ABSTRACT **We present a novel algorithm named FAST for aligning protein three-dimensional structures. FAST uses a directionality-based scoring scheme to compare the intra-molecular residue–residue relationships in two structures. It employs an elimination heuristic to promote sparseness in the residue-pair graph and facilitate the detection of the global optimum. In order to test the overall accuracy of FAST, we determined its sensitivity and specificity with the SCOP classification (version 1.61) as the gold standard. FAST achieved higher sensitivities than several existing methods (DaliLite, CE, and K2) at all specificity levels. We also tested FAST against 1033 manually curated alignments in the HOMSTRAD database. The overall agreement was 96%. Close inspection of examples from broad structural classes indicated the high quality of FAST alignments. Moreover, FAST is an order of magnitude faster than other algorithms that attempt to establish residue–residue correspondence. Typical pairwise alignments take FAST less than a second with a Pentium III 1.2GHz CPU. FAST software and a web server are available at http://biowulf.bu.edu/FAST/. Proteins 2005;58:618–627.**

Key words: protein structure alignment; clique detection; clustering-based algorithm; sensitivity; specificity; alignment quality

INTRODUCTION

The rapid accumulation of protein three-dimensional (3D) structures in the Protein Data Bank[1] (http://www.rcsb.org/pdb/) has made structure comparison an indispensable bioinformatics tool for studying protein function and evolution.[2,3] The goal of structure comparison is to relate proteins based on their structural similarity. It is well known that remote protein homologs may have unrecognizable sequence similarity while their 3D structures are still highly conserved.[4,5] Thus structure alignment algorithms can detect equivalencies that purely sequence-based methods can not.

Many approaches to structural comparison have been proposed, roughly falling into two categories: superposition and clustering.[6] All methods must identify equivalent residue pairs in two proteins; however, methods in the two categories differ in their approaches to quantifying the equivalency. Superposition-based algorithms translate and rotate one protein structure in the 3D space to minimize its *intermolecular* distance to another protein structure.[7–10] Clustering-based methods, in contrast, compare and cluster *intramolecular* residue–residue distances in one protein with those in another protein. Various heuristics have been developed for the latter approach,[11–14] because the general problem is NP-hard.[15]

Despite the plethora of existing algorithms, the protein structure alignment problem is not yet solved. Gerstein and Levitt reported that structure alignments frequently lacked convergence.[8] Finding biologically relevant similarities remains another challenge. To tackle the substantial computational complexity, "slow and reliable" algorithms establish residue–residue correspondence between two structures, while "quick and dirty" algorithms supply coarse alignments using Secondary Structure Elements (SSEs; namely α-helices and β-strands). Some algorithms, in an effort to narrow down the search space and speed up residue-level alignments, first perform SSE alignments to filter out dissimilar proteins.[2,16–18] Unfortunately, this approach suffers from the relatively low accuracy of SSE alignments and in some cases the failure to produce an SSE alignment due to the lack of SSEs in the input structures.

Here we report a novel structure alignment algorithm named FAST (a recursive acronym of FAST Alignment and Search Tool). Because it is extremely difficult to recover from inaccurate SSE alignments, we decided to work directly with protein backbones represented as chains of C_α atoms. According to the classification by Eidhammer et al.,[6] FAST is a clustering-based algorithm. Unlike previous methods in this category, which cluster compatible residue-pairs into candidate alignments, FAST employs various novel techniques to eliminate incompatible residue-pairs and reduce computational complexity. Although FAST does not explicitly assign secondary structures, it detects local backbone geometry and produces biologically meaningful alignments. We evaluated FAST's performance in two ways. We measured its overall sensitivity and specificity of determining whether two proteins can be aligned using the manual SCOP classification of protein folds.[19] For the accuracy of detailed alignments, we tested FAST against all curated structural alignments in HOM-

Grant sponsor: the National Science Foundation; Grant number: DBI-0078194.

*Correspondence to: Zhiping Weng, Biomedical Engineering Department, Boston University, 44 Cummington Street, Boston, MA 02215. E-mail: zhiping@bu.edu

Received 26 February 2004; Accepted 28 July 2004

Published online 17 December 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.20331

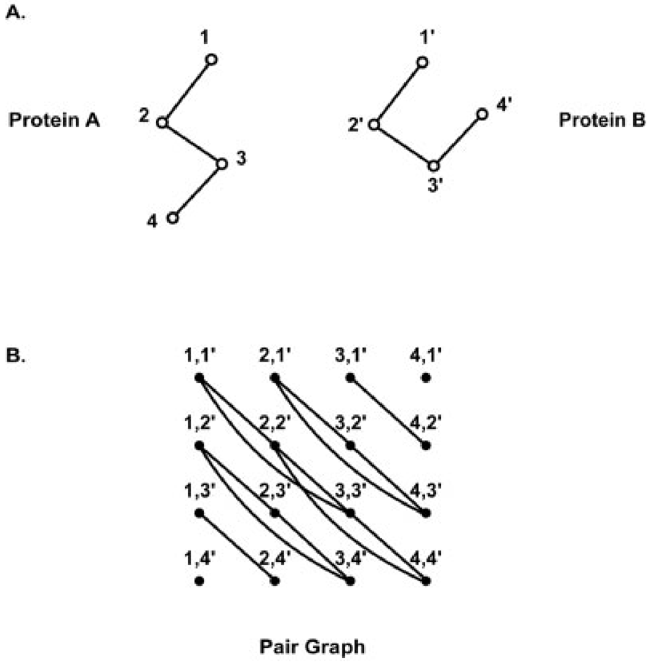

Fig. 1. Schematic illustration of *pair graph* for structure alignment. **A:** Two proteins each with four residues. Each residue is represented by its C_α atom (open circle), and a simplified bond connecting neighboring C_α atoms. **B:** The pair graph corresponding to the two proteins in A. Each vertex indicates a residue pair. (1,1′) indicates the pairing of residue 1 of protein A with residue 1′ in protein B, and so on. Each edge connects two compatible residue pairs. For example, the edge connecting (2,1′) and (4,3′) indicate that the Euclidian distance between residues 2 and 4 in protein A is of similar length to that between residues 1′ and 3′ in protein B. Cliques in the pair graph indicate alignments. There are six cliques in the graph. For example, clique [(2,1′), (3,2′), (4,3′)] indicates the alignment between residues 2, 3, 4 of protein A and residues 1′, 2′, 3′ of protein B.

STRAD.[20] Both comparisons illustrated FAST's superior performance over several other methods.

FAST, as its name suggests, is especially appealing because of its speed. On a Linux computer with a Pentium III 1.2 GHz CPU, a typical pair-wise alignment takes less than a second, an order of magnitude faster than all other residue-level alignment methods that we are aware of. Thus, FAST is a promising tool for scanning large structure databases to detect remote homologs that are elusive for sequence search tools. In this paper we focus on its algorithmic details and alignment quality, and leave a full-fledged structure search engine for the topic of future articles.

METHODS

Optimization Strategy

The problem of searching for the optimal alignment between two protein structures can be formulated as finding the maximal clique in a *pair graph* (Fig. 1), in which vertices represent possible pairings between C_α atoms from the two structures and edges denote compatibility between such pairs. Figure 1(B) illustrates the pair graph for proteins A and B in Figure 1(A), each with four residues. In this graph, an edge is drawn if the difference between the intra-molecular distances between two residue pairs is smaller than a cutoff. Compatibility can also be defined in many other ways and weights can be assigned to edges to signify the degrees of compatibility.[6] There are six cliques in Figure 1(B), each representing a possible structural alignment. We hereafter denote the pair graph with *G(V,E)* where *V* and *E* are vertex and edge sets respectively. We also use the terms *vertex* and *pair* interchangeably. A collection of vertices in *E* can correspond to an alignment if the residue indices within a protein are not reused (because one residue in one protein can only be aligned with one residue in the other protein). If sequential constraint is also imposed, all residue indices of one protein must be in ascending (or descending) order after the residue indices of the other protein have been sorted accordingly. The six cliques in Figure 1(B) satisfy both requirements.

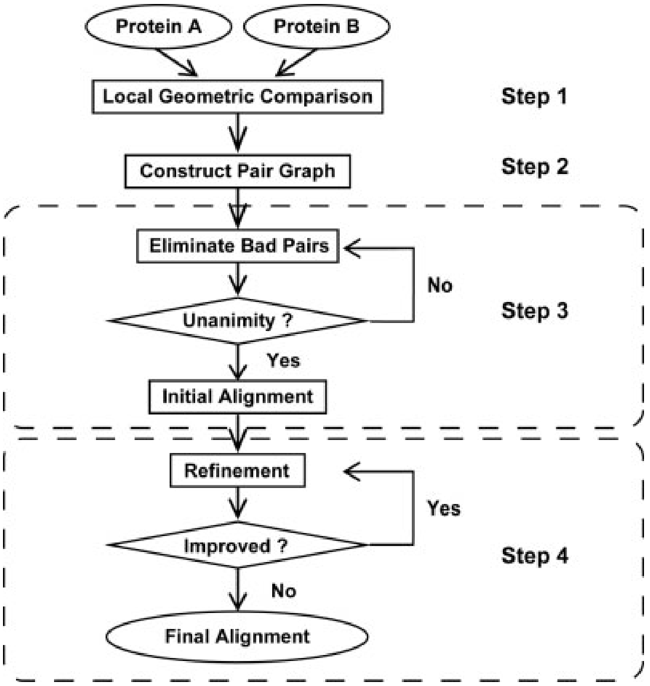

Fig. 2. The flow chart of FAST. The algorithm contains four steps. See the Methods section for more details.

It is well-established that the maximum clique detection problem is NP-hard.[6] For protein structure comparison, the situation is particularly severe due to the large size of the pair graph. For two proteins with *M* and *N* residues respectively, the number of vertices in the pair graph is *MN*, which is in the order of 10^5 for average-size proteins. To tackle this problem, we use a combination of empirical rules to reduce the size and density of *G(V,E)*, until a reasonable approximation is possible.

FAST comprises four steps (Fig. 2). First, we compare the local geometric properties of the two proteins and select a small subset of *MN* pairs as the vertex set to construct *G(V,E)*. Second, we assign edges by comparing intra-molecular relationships, using a directionality-based scoring scheme that promotes sparseness of the graph.

Third, we iteratively prune the graph to eliminate "bad vertices," which are residue pairs that are unlikely to constitute the global optimal alignment, offering the correct alignment a better chance to survive. With the substantially simplified product graph, an initial alignment is easily detected using dynamic programming. Finally, we fine-tune the initial alignment by finding additional equivalent pairs and eliminating bad pairs.

Step 1: Local Geometric Comparison

Local geometric comparison serves as a filter for finding residue pairs. It is performed with a similarity function of two five-residue segments. L_{ij} denotes the similarity between a segment centered around residue i of protein A and a segment centered around residue j in protein B:

$$L_{ij} = \min\,\{D(d_{i-2,i+2}{}^{A}, d_{j-2,j+2}{}^{B}), D(d_{i-2,\,i+1}{}^{A}, d_{j-2,j+1}{}^{B}), D(d_{i-1,\,i+2}{}^{A}, d_{j-1,j+2}{}^{B})\}, \quad (1)$$

$$\text{where } D(d_1, d_2) = 0.1 - |d_1 - d_2| \,/\, (d_1 + d_2).$$

$d_{i-2,i+2}{}^{A}$ indicates the Euclidian distance between C_α atoms $i-2$ and $i+2$ in protein A and likewise for five other such distances [they are indicated with dashed lines in Figure 3(A)]. The first two and last two residues of a protein chain are left out of the calculation because they do not have two neighboring residues on each side.

We then prune $G(V,E)$ in favor of consecutive and high-scoring segments: (a) Pairs with negative L scores are eliminated. (b) Pairs with isolated high L scores, i.e., those that cannot form a stretch of four high-scoring pairs, are also eliminated.

Typically more than 90% pairs are purged without substantially affecting the correct alignment. Even though we do not explicitly consider secondary structures, a residue in an α helix would never be paired with a residue in a β strand because of their distinct local geometric patterns. Thus, FAST implicitly accounts for secondary structure information.

Step 2: Scoring Scheme for Edge Computation

The edge in $G(V,E)$ connecting two vertices (i,j) and (m,n) is assigned the following weight:

$$e_{ij;mn} = (1 - \max\,\{k_d \cdot t_d, k_\alpha \cdot t_\alpha, k_\beta \cdot t_\beta, k_\gamma \cdot t_\gamma\}) \cdot \exp\left(-\left(\frac{d_{i,m}{}^{A} + d_{j,n}{}^{B}}{2d_0}\right)^2\right), \quad (2)$$

$$\text{where } t_d = |d_{i,m}{}^{A} - d_{j,n}{}^{B}|/(d_{i,m}{}^{A} + d_{j,n}{}^{B}),$$

$$t_\alpha = \max\{|\alpha_{\alpha 1} - \alpha'_1|, |\alpha_{-1} - \alpha'_{-1}|\},$$

$$t_\beta = \max\{|\beta_1 - \beta'_1|, |\beta_{-1} - \beta'_{-1}|\},$$

$$t_\gamma = \max\{|\gamma_1 - \gamma'_1|, |\gamma_{-1} - \gamma'_{-1}|\},$$

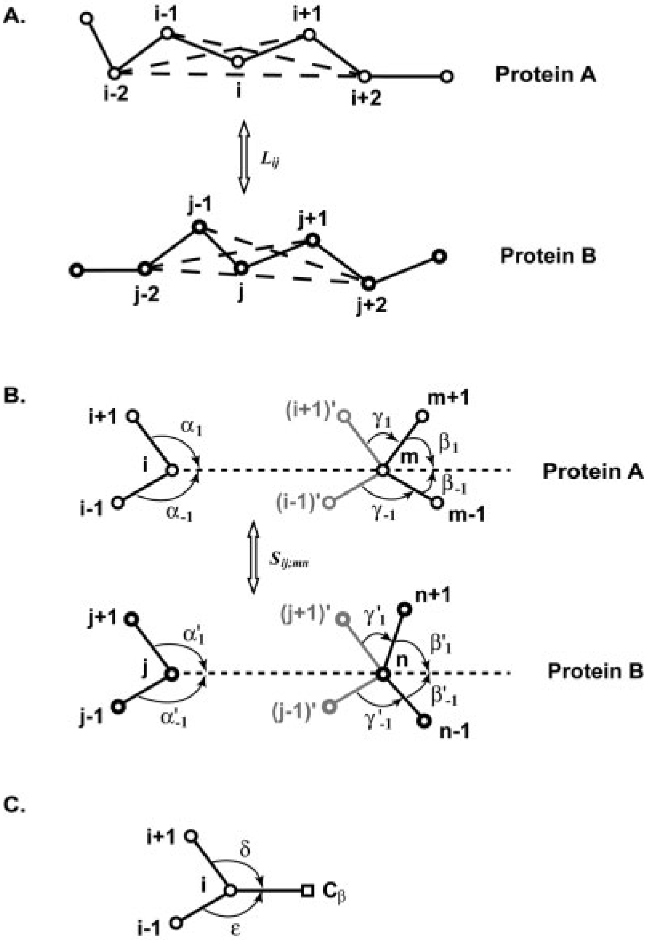

Fig. 3. FAST scoring scheme. Each residue is represented by its C_α atom (open circle). Solid lines connect neighboring C_α atoms. **A:** Local geometric compatibility is defined as a function of six Euclidian distances (Equation 1). These distances are indicated with dashed lines in this figure. **B:** The weight of an edge is defined as a function of two Euclidian distances and twelve angles (Equation 2). The dashed lines in the figure correspond to the two distances. They are extended to facilitate the illustration of the angles. α_1 is the angle between the vector that connects C_α atoms *i* and *i*+1 and the vector that connects C_α atoms *i* and *m*. Other α and β angles are defined similarly. Angle γ_1 is defined between the vector that connects C_α atoms *m* and *m*+1 and the vector that connects C_α atoms *m* and (*i*+1)′, with the latter being the projection of atom *i*+1 such that the *m*-(*i*+1)′ vector is parallel to the *i*-(*i*+1) vector. **C:** Side-chain orientation defined by angles δ and ϵ. C_α atoms are indicated with open circles and the C_β atom of residue *i* is indicated with an open square.

As in Equation 1, $d_{i,m}{}^{A}$ in Equation 2 indicates the Euclidian distance between C_α atoms i and m in protein A and $d_{j,n}{}^{B}$ between C_α atoms j and n in protein B. The distance term t_d and the exponential decay envelope are adopted from the elastic score by Holm and Sander.[13] We introduce three additional terms, t_α, t_β and t_γ, to measure similarity in backbone directionality, described with six angles α_1, β_1, γ_1, α_{-1}, β_{-1} and γ_{-1} in Figure 3(B). k_d, k_α, k_β, and k_γ are empirically determined scaling factors (k_d = 10 Å, $k_\alpha = k_\beta = 4/\pi$ rad and $k\gamma = 3/\pi$ rad).

Careful considerations went into the directionality terms in Equation 2. Congruence of backbone directions is frequently observed in structurally conserved regions, due to the regularity of SSEs. With directionality constraints, our scoring scheme substantially reduces the density of $G(V,E)$. In addition, backbone directionality correlates well with side-chain orientation. As illustrated in Figure 3(C), the

TABLE I. Correlation Between Backbone Direction and Side-Chain Orientation

Angles[a]	δ	ε
Mean	111.15°	121.47°
Standard Deviation	1.27°	0.80°

[a] Angles δ and ε characterize side-chain orientation, as defined in Figure 3(C). Statistics are obtained with all nonterminal nonglycine residues in 4772 SCOP structures sharing less than 40% sequence identity.

vector connecting the C_α and C_β atoms of a residue can be defined with two angles δ and ϵ. Table I indicates that δ and ϵ are highly invariant among all nonterminal, nonglycine residues in 4772 SCOP structures sharing less than 40% sequence identity. Gerstein and Levitt reported that C_β coordinates were needed for correctly aligning some difficult cases.[8] Our directionality terms implicitly take side-chain orientations into account.

With the above defined vertices and edges for *G(V,E)*, the total similarity score for alignment *X* is

$$S_X = \sum_{(i,j)eX;(m,n)eX;(m,n)\neq(i,j)} e_{ij;mn} \qquad (3)$$

and the optimal alignment *X** should achieve the maximal similarity score.

Step 3: Further Pruning and Initial Alignment

An edge with a positive weight supports the coexistence of two residue pairs in some alignment. If we sum all positive weights of edges associated with vertex (*i,j*), and call this the total score (*T*) of the vertex (Equation 4), we would expect *T* to be high if (*i,j*) is contained in the optimal alignment *X**.

$$T_{ij} = \sum_{(m,n)eV} \max\{e_{ij;mn},0\} \qquad (4)$$

Unfortunately the converse is not necessarily true, because all alignments that contain (*i,j*) contribute to *T*. We have devised a strategy to iteratively eliminate vertices that are most unlikely to be part of *X**. Without the contributions from such bad vertices, hopefully *X** becomes apparent in the *T* matrix. The success of this strategy relies on a sparse graph. Three empirical rules are used to define bad pairs: (a) A vertex receiving a low *T* score is eliminated. The actual threshold is given by a percent cutoff. The default is to remove 10% low-T vertices in each row and each column of the T_{ij} matrix. (b) If the *T* score of a vertex is due to scattered contributors that do not form stretches, the vertex is eliminated. (c) If the two residues of a vertex are isolated, i.e., their neighboring residues are no longer in *G(V,E)*, the vertex is eliminated.

In order to measure the extent to which the graph is close to a clique, we define the *degree of unanimity* of *G(V,E)* as the number of edges with positive weights divided by the total number of possible edges. We expect the degree of unanimity to increase as we iteratively eliminate more bad vertices. The process terminates when there is no further improvement in unanimity. There are two possibilities at this point: (a) All singletons and insignificant local maxima have been eliminated and the global optimum is dominant. (b) Sometimes a tie is reached among multiple comparable maxima. This is often the case when one or both structures contain repeats.

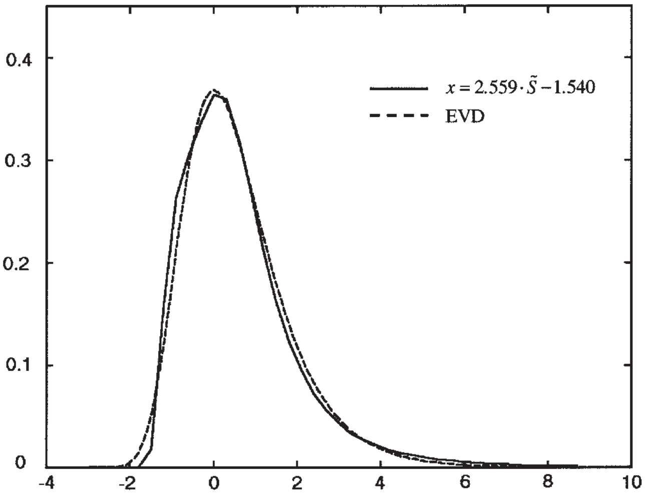

Fig. 4. The fit of an Extreme value distribution (EVD; dashed line) to normalized scores $\tilde{S}$ (solid line). It is obtained by aligning 1,000,000 pairs of dissimilar structures (different SCOP folds). The dashed line is the standard EVD with probability density function $p(x) = e^{-x} \cdot e^{-e^{-x}}$. The distribution of $\tilde{S}$ is best fitted to EVD by the relationship $x = 2.559{*}\tilde{S} - 1.540$.

For case (a) an initial alignment X^0 can be easily identified by running dynamic programming[21] with T_{ij} as the similarity score and zero gap penalty. For case (b), which is often reflected by a low degree of unanimity that cannot be improved by further pruning, we must break the tie to retrieve the most significant alignment. Because the search space is sufficiently reduced at this point, we split the graph using a vertex with large *T* as the pivot. Vertices incompatible to the pivoting vertex are discarded. The remaining sub-graph is then subjected to further pruning until case (a) is reached. At most three pivots are tried and the best alignment is kept.

Step 4: Alignment Refinement

The initial alignment X^0 may contain residue pairs that do not belong to the optimal alignment and some biologically relevant pairs may be dropped during pruning. We try to improve X^0 by recovering additional relevant pairs and removing bad pairs. Given X^0, the compatibility of a residue pair (*i,j*) is defined as:

$$R_{ij} = \sum_{(m,n)eX^0;\ (m,n)\neq(i,j)} e_{ij;mn} \qquad (5)$$

The first round of improved alignment X^1 can be constructed by running dynamic programming[21] with R_{ij} as the similarity score and zero gap penalty. The process is performed for up to five rounds, with a new *R* matrix and updated alignment constructed each time, or until the alignment converges. Obviously, the total similarity score upon convergence is exactly the target function defined in

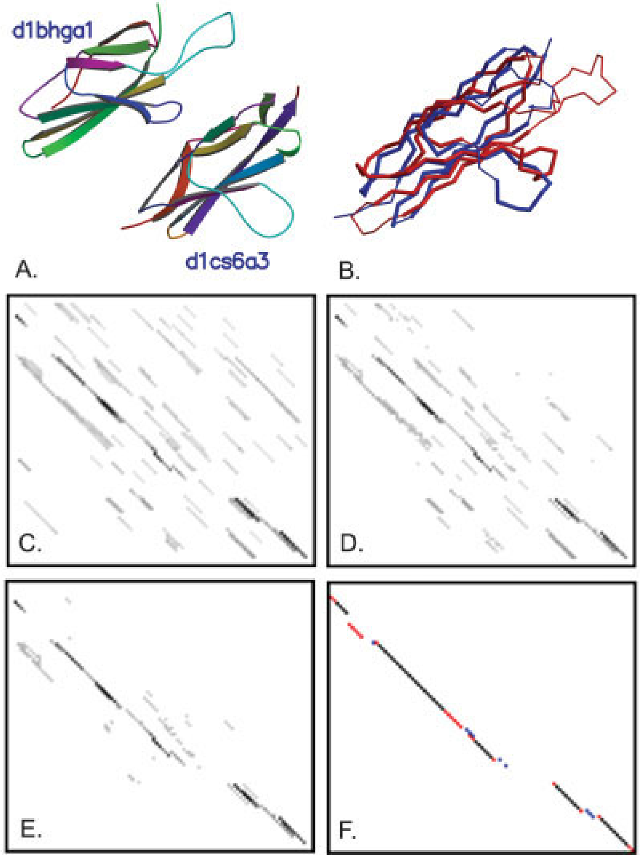

Fig. 5. Aligning d1bhga1 and d1cs6a3. **A:** The 3D structures of the two proteins with β-strands indicated in block arrows. **B:** Superposed backbones with aligned regions highlighted in thick lines, d1bhga1 in red and d1cs6a3 in blue. A and B were generated with MOLSCRIPT[33] and RASTER3D.[34] **C:** The first *T* score matrix (Equation 4) following local geometry comparison. Protein d1bhga1 (103 residues) is placed vertically and d1cs6a3 (91 residues) is placed horizontally. The intensity of each pixel represents the *T* score of the corresponding residue pair. The pixels with *T* scores lower than the default cutoff are omitted. **D:** The *T* score matrix after three rounds of pruning (see Step 3 of the FAST algorithm in the Methods section). **E:** The *T* score matrix after nine rounds of pruning. One alignment becomes dominant. **F:** The initial alignment (blue and black pixels) and the final refined alignment (red and black pixels).

Equation 3. As the final step, the alignment is trimmed by discarding pairs that do not form a stretch of four or more aligned residue pairs.

Score Normalization and Statistical Significance

The *raw score* defined by Equation 3 indicates the overall similarity between two protein structures. Although capable of capturing the optimal alignment, it is not ideal for determining whether an alignment is significant, because large proteins are more likely to produce a high-scoring alignment purely by chance. We use a simple normalization scheme:

$$\tilde{S} = \frac{S_X}{\sqrt{M \cdot N}}, \tag{6}$$

where M and N are numbers of residues in the two proteins. The normalized score $\tilde{S}$ approximately follows an extreme value distribution (EVD), which allows us to compute statistical significance in a way similar to the BLAST P-value.[22] Figure 4 indicates the good fit of an EVD to the $\tilde{S}$ scores of aligning 1,000,000 pairs of dissimilar structures.

TABLE II. Sensitivities at Various Specificity Cutoffs

Number of false positives (Specificity)	Number of true positives (Sensitivity)			
	FAST	DaliLite	K2	CE
1500	62979	57605	49073	33654
(99.5%)	(61.70%)	(56.43%)	(48.07%)	(32.97%)
3000	72443	67060	56704	41504
(99%)	(70.97%)	(65.70%)	(55.55%)	(40.66%)
6000	80083	75816	63536	50981
(98%)	(78.45%)	(74.27%)	(62.24%)	(49.94%)
15000	87300	84505	71308	65274
(95%)	(85.52%)	(82.79%)	(69.86%)	(63.95%)
30000	91241	88905	76404	75659
(90%)	(89.38%)	(87.10%)	(74.85%)	(74.12%)
60000	94684	92085	80690	85424
(80%)	(92.76%)	(90.21%)	(79.05%)	(83.69%)

Algorithm Implementation and Availability

FAST was implemented in ANSI C on a personal computer running Linux and has been ported to several other platforms including SGI/IRIX, Alpha64 and IBM SP. It is self-contained and fully automated. For consistency and simplicity, we do not allow any case-by-case parameter modification. A single set of built-in parameters is used throughout. FAST is freely available to academic users at http://biowulf.bu.edu/FAST/. A submission form is also provided for pair-wise alignment over the Web.

RESULTS AND DISCUSSION

An Example

Figure 5 illustrates the process of aligning d1bhga1 (PDB code 1BHG, A chain, residues 226-328) and d1cs6a3 (PDB code 1CS6, A chain, residues 209–299), both belonging to the immunoglobulin-like β-sandwich SCOP fold.[19] Structures in this fold are challenging for alignment algorithms because of the large number of local maxima (the two β-sheets can be aligned either way and β-strands can slide against each other, resulting in many high-scoring alignments). Furthermore, this fold can accommodate many sequences as well as great structural variation. The optimal alignment between d1bhga1 and d1cs6a3 [Fig. 5(B)] shows high variation in loops and the ends, and some variation even in the core of the fold.

Figure 5(C) indicates the T matrix (103 by 91 residues) at the beginning of Step 3 (Methods). There are many high-scoring diagonals in Figure 5(C), indicating the large number of local maxima. The T matrices after three and nine rounds of pruning are shown in Figures 5(D) and 5(E) respectively. All local maxima have been eliminated by round 9, and only one alignment is dominant [Figure 5(E)]. This alignment indeed corresponds to the global optimal alignment [Fig. 5(F)].

TABLE III. Total/Average Running Time on the SCOP Data Sets (in seconds)[†]

Method	FAST	DaliLite	K2	CE
Positive Set (102,077 pairs)	53760/0.53	1360140/13.32	798660/7.82	337920/3.31
Negative Set (300,000 pairs)	63240/0.21	2093520/6.98	300480/1.00	310380/1.03

[†]Running time is measured by the wall time used by a Pentium III 1.2 GHz CPU.

This example illustrates that the multiple-step pruning heuristic of FAST is highly effective in removing the residue pairs that are not part of the global optimum. The resulting pair graph is sparse and therefore it is relatively easy and rapid to detect the correct alignment. This alignment takes only 0.11 seconds using a single Pentium III 1.2GHz CPU.

Overall Accuracy

Gerstein and Levitt pointed out the importance of using manual standards to assess the performance of automated alignment algorithms.[8] They tested their algorithm against 2107 pairs of SCOP proteins. We shared the same philosophy but switched to a much larger data set: SCOP version 1.61[19] that covered the entire Protein Data Bank.[1] To test FAST on the most difficult cases, we chose a nonredundant set of 4772 structures sharing less than 40% sequence identity as provided by the ASTRAL compendium.[23] These were all single-domain proteins defined by SCOP, with an average length of 181 amino acids. We determined the sensitivities at various specificities using all 102,077 same-fold pairs as the positive set and 300,000 randomly chosen different-fold pairs as the negative set.

For comparison we ran several other structure alignment algorithms on the same dataset. There were two limiting factors for selecting other algorithms: the availability of the software, and the speed of the software to cope with the large data set. Our final list included DaliLite,[24] CE,[25] and K2.[17,18] DaliLite (an offline version of DALI) is a branch-and-bound clustering-based algorithm. A number of approximations are introduced to cope with the combinatorial complexity and Monte Carlo optimization is performed to refine the alignment in the entire search space. CE builds structure alignment by incremental combinational extension of optimal paths. A list of aligned fragment pairs is first identified. The optimal alignment path is built up by combining aligned pieces, followed by a dynamic programming refinement. K2, developed earlier by our lab, exemplifies a hierarchical alignment strategy. Because proteins are most conserved in cores as represented by secondary structures, K2 first generates an SSE alignment. A genetic algorithm is then used to optimize the alignment at the residue level.

Table II summarizes the comparison of FAST, DaliLite, CE, and K2. FAST and DaliLite are more accurate than the other two methods. At reasonable specificities (95–99%), the overall sensitivity of FAST is 3–5% above DaliLite and more than 10% higher than K2 and CE. Analysis of false-negatives indicated that the major issue of K2 resided in the SSE alignment stage. The quality of SSE alignment was limited by a number of factors, such as the accuracy of SSE assignment and the availability of sufficient SSEs to work with. As a result, K2 had difficulty with small proteins and structures with low SSE content. To avoid unreliable SSE alignments was in fact one of our motivations for developing the FAST algorithm.

Even at the low specificity of 80%, the two best methods, FAST and DaliLite, could not detect all pairs of structures designated by SCOP to be in the same fold (the sensitivities were 93% and 90% for FAST and DaliLite respectively). We therefore decided to investigate the worst cases, referred to as same-fold pairs that were assigned extremely poor scores by one or both of these two programs. The cutoff we chose for DaliLite was Z-score = 0 (i.e., no alignment generated at all), which corresponded to a specificity of 67%. DaliLite assigned zero Z-scores to 8056 same-fold pairs. FAST computed $\tilde{S}$ below the equivalent threshold of 0.881 for 5098 same-fold pairs. We asked whether one method could correct the errors of the other method by assigning a score above the method's 95% specificity cutoff. We found that 2242 of the worst 8056 same-fold pairs defined by DaliLite were corrected by FAST and 783 of the 5098 FAST cases were corrected by DaliLite. Therefore, the two methods complemented each other in coping with some of the most difficult cases.

Some difficult cases reported by DaliLite were due to problematic structures. Among the 4772 structures, 37 had multiple chains and 12 had C_α atoms only. DaliLite failed on all cases that involved one of these 49 structures. This accounted for 1252 out of the 102,077 same-fold pairs and 6008 out of the 300,000 different-fold pairs. The 8056 worst cases include all 1252 cases, out of which 977 were corrected by FAST, indicating that problematic input structures had little impact on FAST's performance.

Both DaliLite and FAST failed on 3081 (approximately 3%) extremely difficult cases. Small proteins appeared to be a major problem. There were 1944 or about two-thirds of the above extremely difficult pairs between proteins shorter than 80 residues. Others involved proteins of permuted SSEs or different topologies. One example is d1byi__ vs. d1qhla_, both belonging to the "P-loop containing nucleotide triphosphate hydrolases" fold. However, the β sheet in d1byi__ is parallel and that in d1qha_ is anti-parallel. Both FAST and DaliLite enforce strict sequential constraints and are thus unable to detect such similarities.

False positives by FAST were mostly due to common substructures shared by proteins with different overall structures. Others represented surprising similarities across different SCOP folds. There were numerous examples among the top 200 false positives. The example below is not chosen from top 200 false positives; however, the similarity is still remarkable. SCOP entry d1e79d2 (PDB code 1E79, chain D, residues 9–81) belongs to the "domain of α and β subunits of F1 ATP synthase-like" fold (SCOP ID b.49.1.1, described as barrel, closed; n = 6, S = 8; Greek-key). Its structure is superposable on SCOP structure d1g7sa1 (PDB code 1G7S, chain A, residues 228–328), which belongs to the "reductase/isomerase/elongation-factor common domain" fold (SCOP ID b.43.3.1, described as barrel, closed; n = 6, S = 10; Greek-key).

Computational Efficiency

The SCOP data set (402,077 pair-wise alignments) requires considerable computing power. We carried out the calculation in parallel using a Linux computer cluster that consisted of 128 computing nodes, each with two 1.2 GHz Pentium III CPUs and 4 GB RAM. The entire data set was divided into 84 groups and assigned to 42 compute nodes. Table III compares the speeds of the four methods. We measure running time for the positive set and the negative set separately because it heavily depends on the average number of aligned residues. On both data sets FAST was much faster than other methods, about 30 times faster than DaliLite and 5–10 times faster than K2 and CE. The average time for aligning a pair of similar/dissimilar structures using FAST was 0.51/0.21 seconds.

Alignment Quality

For an automated structure alignment algorithm, it is important to investigate whether it produces biologically meaningful alignments. This is especially important for FAST because we do not use any secondary structure information, or perform rigid-body superposition of the two structures. Generating high-quality alignments is a challenging problem and biologically meaningful alignments are often barely distinguishable from local maxima in terms of alignment length or the root-mean-square deviation (RMSD) of the structures superposed at aligned residues.[26]

We compared automated alignments by FAST with manually curated alignments in the HOMSTRAD database.[20] As of March 2003, HOMSTRAD contained annotated alignments of 1033 homologous protein families. For each family in the database, we ran FAST between each pair of proteins and tallied matches and discrepancies compared to the corresponding manual alignment. In aligned regions, FAST agreed with HOMSTRAD for 96% residue pairs. In particular, they were in perfect agreement for 332 out of the 1033 families. Thus, FAST reliably generates biologically correct residue pairs in a human expert's point of view. Detailed analysis identified two types of discrepancies: (1) most mismatches resided in loop regions where structures were highly variable; (2) misalignments in cores were usually characterized by shifts in SSEs.

We provide examples across broad structural classes in Table IV. The FAST alignments invariably had fewer residue pairs and lower RMSDs than the corresponding manual alignments. On average, FAST alignments were 7% shorter than the corresponding HOMSTRAD align-

TABLE IV. Comparison of FATCAT, DALI, FAST, and HOMSTRAD Alignments

			HOMSTRAD[c]		FATCAT[d]			DALI[e]			FAST[f]		
Proteins[a]		Structure class[b]	Length	RMSD	Length	Match	RMSD	Length	Match	RMSD	Length	Match	RMSD
1df4a 3–64	1qcea 1–123	All α	57(21)	2.5	57(8)	0%	2.1	57(2)	0%	1.5	55(29)	55%	1.2
1hx8a 22–299	1hg5a 19–281	All α	258(173)	1.1	258(172)	100%	1.1	258(173)	99%	1.1	255(170)	99%	1.1
2ahja 4–203	1irea 2–204	α + β	192(86)	4.3	186(83)	98%	1.1	189(83)	97%	2.0	187(82)	89%	2.0
1h7sa 232–364	1b63a 217–331	α + β	105(15)	2.2	105(15)	97%	2.2	105(15)	97%	2.2	98(15)	99%	2.0
1ed9a 1–449	1ew2a 1–479	α/β	403(127)	5.6	395(122)	87%	3.0	376(124)	90%	1.9	343(119)	98%	1.7
1oyc 1–339	2tmda 1–340	α/β Barrel	330(81)	3.6	333(77)	84%	3.1	323(81)	88%	2.6	284(77)	97%	2.3
1fjna 1–39	1ica 1–40	Small Protein	33(13)	4.7	30(11)	93%	2.1	29(10)	83%	1.9	28(12)	100%	1.9
1tpn 1–50	1fbr 1–46	Small Disulphide	43(12)	2.4	43(11)	98%	2.1	43(11)	98%	2.0	40(10)	93%	2.2
1el2a 24–262	1c3wa 5–231	Membrane All α	220(75)	1.7	222(73)	95%	1.8	220(75)	96%	1.6	214(74)	97%	1.5
1af6a 1–421	1a0t 71–483	Membrane All β	377(102)	4.6	379(96)	91%	3.1	367(95)	91%	2.5	323(86)	97%	1.8
1hc1 5–653	1lla 2–628	Multi-domain	582(200)	2.3	585(199)	95%	2.1	575(201)	96%	1.8	546(198)	97%	1.7

[a] The first four characters indicate the PDB code of the structure. The fifth character indicates the chain ID whenever available. The range of residues is indicated after each PDB code.

[b] Structural classes are defined according to SCOP.

[c] Length indicates the number of aligned residues according to HOMSTRAD. Values in parentheses indicate numbers of identities in HOMSTRAD alignments. RMSD indicates the root-mean-square deviation of aligned residues (according to HOMSTRAD) after superposition, obtained with a least-square fitting algorithm.[35] All RMSD values are in Å.

[d] Length indicates the number of aligned residues according to FATCAT, with the number of identities indicated in parentheses. Match indicates the percentage of aligned residues according to FATCAT that are also deemed equivalent by HOMSTRAD. RMSD indicates the root-mean-square deviation of aligned residues according to FAST after superposition. FATCAT alignments were produced using the FATCAT web-server (http://fatcat.ljcrf.edu/fatcat/).

[e] Similarly for DALI alignments. We used the DALI mail-server (http://www.ebi.ac.uk/dali/Interactive.html) to produce these alignments.

[f] Similarly for FAST alignments.

A.

HOMSTRAD Alignment:

```
 2ahja:      I-------DHPAQAPVSDRAWALFRALDGKGLVPDGYVEGWKKTFEEDFSPRRGAELVARAWTDPEFRQL
 1irea:      TENILRKSDEEIQKEITARVKALESMLIEQGILTTSMIDRMAEIYENEVGPHLGAKVVVKAWTDPEFKKR

 2ahja:      LLTDGTAAVAQYGYLGPQGEYIVAVEDTPTLKNVIVCSLSTAWPILGLPPTWYKSFEYRARVVREPRKVL
 1irea:      LLADGTEACKELGIGGLQGEDMMWVENTDEVHHVVVCTLSYPWPVLGLPPNWFKEPQYRSRVVREPRQLL

 2ahja:      SEM-GTEIASDIEIRVYDTTAETRYMVLPQRPAGTEGWSQEQLQEIVTKDCLIGVAIP-QV
 1irea:      KEEFGFEVPPSKEIKVWDSSSEMRFVVLPQRPAGTDGWSEEELATLVTRESMIGVEPAKAV
```

FAST Alignment:

```
 2ahja:      IDHP----AQAP---VSDRAWALFRALDGK----GLVPDGYVEGWKKTFEEDFSPRRGAELVARAWTDPE
 1irea:      ----TENILRKSDEEIQKEITARVKALESMLIEQGILTTSMIDRMAEIYENEVGPHLGAKVVVKAWTDPE

 2ahja:      FRQLLLTDGTAAVAQYGYLGPQGEYIVAVEDTPTLKNVIVCSLSTAWPILGLPPTWYKSFEYRARVVREP
 1irea:      FKKRLLADGTEACKELGIGGLQGEDMMWVENTDEVHHVVVCTLSYPWPVLGLPPNWFKEPQYRSRVVREP

 2ahja:      RKVLSE-MGTEIASDIEIRVYDTTAETRYMVLPQRPAGTEGWSQEQLQEIVTKDCLIGVAIPQV--
 1irea:      RQLLKEEFGFEVPPSKEIKVWDSSSEMRFVVLPQRPAGTDGWSEEELATLVTRESMIGVEPAK-AV
```

B.

HOMSTRAD Alignment:

```
 1df4a:      ---------IVQQQNNLLRAIEAQQHLLQLTVWGIKQLQAG-----------------------------
 1qcea:      AQSRTLLAGIVQQQQQLLDVVKRQQELLRLTVWGTKNLQTRVTAIEKYLKDQAQLNAWGAAFRQVAHTTV

 1df4a:      -----------GWMEWDREINNYTSLIHSLIEESQN-----------------
 1qcea:      PWPNASLTPKWNNETWQEWERKVDFLEENITALLEEAQIQQEKNMYELQKLNS
```

FAST Alignment:

```
 1df4a:      ---------IVQQQNNLLRAIEAQQHLLQLTVWGIKQLQA------------------------------
 1qcea:      AQSRTLLAGIVQQQQQLLDVVKRQQELLRLTVWGTKNLQ-TRVTAIEKYLKDQAQLNAWGAAFRQVAHTT

 1df4a:      --------------GGWMEWDREINNYTSLIHSLIEESQN---------------
 1qcea:      VPWPNASLTPKWNNETWQEWERKVDFLEENITALLEEAQ-IQQEKNMYELQKLNS
```

Fig. 6. Examples of discrepancy between FAST and HOMSTRAD (underscored). Aligned positions with identical residue are indicated with bold letters. RMSD and other information of the alignments are listed in Table IV. **A:** $\alpha+\beta$ proteins 2ahja:4-203 and 1irea:2-204. **B:** All-α proteins 1df4a:3-64 and 1qcea:1-123.

ments. Visual inspection revealed that the missing pairs were exclusively in loops and chain termini, where structural variation was so great that FAST left residues unaligned to preserve maximal structural resemblance in aligned regions. Consequently, FAST alignments had lower RMSD. Thus, such behavior does not indicate inaccuracy of the FAST algorithm.

Our results on aligning SCOP proteins indicated that FAST and DaliLite were more accurate then K2 and CE (see the Overall Accuracy section above). We thus chose to evaluate the quality of DALI alignments for the test cases in Table IV. We also investigated a newly developed algorithm FATCAT,[27] which was capable of incorporating structural flexibility into the alignment. Because none of the other algorithms considered here had this feature, we ran FATCAT in the rigid mode, i.e., without allowing structural flexibility. DALI and FATCAT produced alignments that were of similar lengths to, but with lower RMSDs than the corresponding HOMSTRAD alignment. FAST produced the shortest alignments accompanied by the lowest RMSDs, except for one case (1tpn-1fbr). For the alignment generated by each algorithm, we recorded the percentage of aligned residues that were also deemed aligned by HOMSTRAD (the "Match" column in Table IV). This quantity reflects the overall quality of an alignment. Except for two cases (2ahja-1irea and 1tpn-1fbr), FAST achieved higher match percentages than both DALI and FATCAT. For 1tpn-1fbr, the FAST alignment was only slightly inferior in match percentage (93% vs. 98% for DALI and FATCAT) and visual inspection indicated that all disagreements located in loops. The FAST alignment of 2ahja and 1irea was shifted by one turn in the N-terminal helix, while DALI and FATCAT both agreed with HOMSTRAD [Figure 6(A); Table IV]. Protein 1irea had a long N-terminal α-helix and FAST superposed the shorter helix of 2ahja onto its closest counterpart in the 1irea helix, resulting in the misalignment.

For three cases, FAST produced alignments with substantially higher match percentages than both DALI and FATCAT. The most striking case was 1df4a-1qcea, for which both DALI and FATCAT completely disagreed with HOMSTRAD (thus 0% match percentage), although the RMSDs of their alignments were still small (2.1 Å over 57 residues for FATCAT and 1.5 Å over 57 residues for DALI, compared with 1.2 Å over 55 residues for FAST). FAST's match percentage was also low for this case (55%), which urged us to examine the alignments carefully. Figure 6(B) compares the alignments of FAST and HOMSTRAD, differing by a half helical turn in the underlined region. The FAST alignment is clearly better, indicated by not only a much lower RMSD (Table IV; 1.2 Å over 55 aligned residues, compared with 2.5 Å over 57 aligned residues for HOMSTRAD), but also higher sequence identity [29 identities for FAST and 21 for HOMSTRAD as indicated in Table IV; identical residues in the discrepant region are shown as bold letters in Fig. 6(B)]. DALI and FATCAT have indeed misaligned completely, indicated by poor sequence identities in aligned regions (eight identities for FATCAT and two for DaliLite, compared with 29 for FAST). For the second case (1ed9a-1ew2a), both DALI and FATCAT misaligned a variable N-terminal helix. For the third case (1af6a-1a0t), DaliLite and FATCAT disagreed with HOMSTRAD at several secondary structure regions.

Nonsequential and Flexible Alignments

Among the programs considered here, FAST and CE[25] can only perform sequential alignments while DALI[13] and K2[17,18] are also able to perform nonsequential alignments, although at the expense of substantially greater computational complexity. DaliLite[24] can only perform sequential alignments. If used to align proteins whose equivalent segments conform to different sequential orders, sequentially constrained algorithms can only find the highest-scoring sequential segment.

In this paper, we only consider rigid-body structural alignment algorithms. Other algorithms can take into account structural flexibility, either by partitioning proteins into rigid domains or by introducing hinge regions.[27–31] We investigated the performance of a newly developed flexible-alignment algorithm FATCAT when used in the rigid mode. In terms of alignment quality, FATCAT is slightly inferior to FAST, but comparable to DaliLite. Thus, although FATCAT is an algorithm designed for incorporating structural flexibility, it produces high-quality alignment in the rigid mode.

CONCLUSION

We have developed a new protein structure alignment algorithm FAST. It sets itself apart in a number of ways. First, FAST was carefully designed and implemented to work much more efficiently than three widely used algorithms that produce residue-level alignments. We show that this computational task, previously taking seconds and even minutes, can be reliably accomplished on a sub-second scale. More importantly, the speedup does not lead to loss in accuracy. Large-scale comparison using the SCOP fold classification as the gold standard shows that FAST is more robust than the slower methods, including the most widely used DALI. Unlike some other rapid alignment methods, FAST does not first construct a low-resolution SSE alignment. We work exclusively with backbone C_α atoms; therefore we do not relay on another program (such as DSSP[32]) to properly assign SSEs. The excellent agreement between automated alignments by FAST and manually curated ones in HOMSTRAD supports the validity of our approach. We attribute the success to our scoring scheme that combines C_α–C_α distance and chain directionality. The additional resolving power offered by the directionality terms not only greatly favors biologically important alignment, but also dramatically simplifies the optimization problem.

ACKNOWLEDGMENTS

This work was supported by NSF DBI-0078194. Calculations were performed on a Linux computer cluster supported by NSF MRI DBI-0116574. We thank Joseph Szustakowski for insightful discussions.

REFERENCES

1. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Res 2000;28:235–242.
2. Gibrat JF, Madej T, Bryant SH. Surprising similarities in structure comparison. Curr Opin Struct Biol 1996;6:377–385.
3. Holm L, Sander C. Searching protein structure databases has come of age. Proteins 1994;19:165–173.
4. Chothia C, Lesk AM. The relation between the divergence of sequence and structure in proteins. EMBO J 1986;5:823–826.
5. Murzin AG. How far divergent evolution goes in proteins. Curr Opin Struct Biol 1998;8:380–387.
6. Eidhammer I, Jonassen I, Taylor WR. Structure comparison and structure patterns. J Comput Biol 2000;7:685–716.
7. Falicov A, Cohen FE. A surface of minimum area metric for the structural comparison of proteins. J Mol Biol 1996;258:871–892.
8. Gerstein M, Levitt M. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. Protein Sci 1998;7:445–456.
9. Taylor WR. Protein structure comparison using iterated double dynamic programming. Protein Sci 1999;8:654–665.
10. Jewett AI, Huang CC, Ferrin TE. MINRMS: an efficient algorithm for determining protein structure similarity using root-mean-squared-distance. Bioinformatics 2003;19:625–634.
11. Grindley HM, Artymiuk PJ, Rice DW, Willett P. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. J Mol Biol 1993;229:707–721.
12. Alexandrov NN, Fischer D. Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures. Proteins 1996;25:354–365.
13. Holm L, Sander C. Protein structure comparison by alignment of distance matrices. J Mol Biol 1993;233:123–138.
14. Escalier V, Pothier J, Soldano H, Viari A. Pairwise and multiple identification of three-dimensional common substructures in proteins. J Comput Biol 1998;5:41–56.
15. Lathrop RH. The protein threading problem with sequence amino acid interaction preferences is NP-complete. Protein Eng 1994;7:1059–1068.
16. Singh AP, Brutlag DL. Hierarchical protein structure superposition using both secondary structure and atomic representations. Proc Int Conf Intell Syst Mol Biol 1997;5:284–293.
17. Szustakowski JD, Weng Z. Protein structure alignment using a genetic algorithm. Proteins 2000;38:428–440.
18. Szustakowski JD, Weng Z. K2: protein structure comparisons and their statistical significance. In: Fogel G, Corne D, editors. Evolutionary Computation in Bioinformatics: Morgan Kaufmann; 2002.
19. Lo Conte L, Brenner SE, Hubbard TJ, Chothia C, Murzin AG. SCOP database in 2002: refinements accommodate structural genomics. Nucleic Acids Res 2002;30:264–267.
20. Mizuguchi K, Deane CM, Blundell TL, Overington JP. HOMSTRAD: a database of protein structure alignments for homologous families. Protein Sci 1998;7:2469–2471.
21. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 1970;48:443–453.
22. Karlin S, Altschul SF. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc Natl Acad Sci USA 1990;87:2264–2268.

23. Brenner SE, Koehl P, Levitt M. The ASTRAL compendium for protein structure and sequence analysis. Nucleic Acids Res 2000; 28:254–256.
24. Holm L, Park J. DaliLite workbench for protein structure comparison. Bioinformatics 2000;16(6):566–567.
25. Shindyalov IN, Bourne PE. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng 1998;11:739–747.
26. Godzik A. The structural alignment between two proteins: is there a unique answer? Protein Sci 1996;5:1325–1338.
27. Ye Y, Godzik A. Flexible structure alignment by chaining aligned fragment pairs allowing twists. Bioinformatics 2003;19 Suppl 2:II246–II255.
28. Wriggers W, Schulten K. Protein domain movements: detection of rigid domains and visualization of hinges in comparisons of atomic coordinates. Proteins 1997;29:1–14.
29. Boutonnet NS, Rooman MJ, Ochagavia ME, Richelle J, Wodak SJ. Optimal protein structure alignments by multiple linkage clustering: application to distantly related proteins. Protein Eng 1995;8: 647–662.
30. Ochagavia ME, Richelle J, Wodak SJ. Advanced pairwise structure alignments of proteins and analysis of conformational changes. Bioinformatics 2002;18:637–640.
31. Shatsky M, Nussinov R, Wolfson HJ. Flexible protein alignment and hinge detection. Proteins 2002;48:242–256.
32. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 1983;22:2577–2637.
33. Kraulis PJ. MOLSCRIPT: A program to produce both detailed and schematic plots of protein structures. J Appl Cryst 1991;24:946–950.
34. Merritt EA, Bacon DJ. Raster3D—photorealistic molecular graphics. Methods Enzymol 1997;277:505–524.
35. Kabsch W. A discussion of the solution for the best rotation to relate two sets of vectors. Acta Crystallogr 1978;34:827–828.