Protein Structure Alignment Using a Genetic Algorithm

Joseph D. Szustakowski and Zhiping Weng*
Boston University, Department of Biomedical Engineering, Boston, Massachusetts

ABSTRACT **We have developed a novel, fully automatic method for aligning the three-dimensional structures of two proteins. The basic approach is to first align the proteins' secondary structure elements and then extend the alignment to include any equivalent residues found in loops or turns. The initial secondary structure element alignment is determined by a genetic algorithm. After refinement of the secondary structure element alignment, the protein backbones are superposed and a search is performed to identify any additional equivalent residues in a convergent process. Alignments are evaluated using intramolecular distance matrices. Alignments can be performed with or without sequential connectivity constraints. We have applied the method to proteins from several well-studied families: globins, immunoglobulins, serine proteases, dihydrofolate reductases, and DNA methyltransferases. Agreement with manually curated alignments is excellent. A web-based server and additional supporting information are available at http://engpub1.bu.edu/~josephs. Proteins 2000;38:428–440.** © 2000 Wiley-Liss, Inc.

Key words: protein structure alignment; bioinformatics; protein fold; genetic algorithm; distance matrices

INTRODUCTION

Proteins spontaneously fold into intricate three-dimensional (3D) structures. The advances of techniques such as X-ray crystallography and nuclear magnetic resonance have brought about the determination of more than 10,000 protein structures—a number that is increasing rapidly (http://www.rcsb.org/pdb). Ever since there were more than a handful of 3D structures, comparing or aligning them has been an important technique for elucidating fundamental principles of protein structure, function, and evolution.[1] Structure alignment involves establishing equivalencies between residues in two proteins based on their 3D coordinates. A structure alignment is essentially a list of residue pairs from two proteins that should superpose closely after one protein is translated and/or rotated rigidly.

Generally speaking, structure alignment has two goals. The first is to determine whether two proteins share the same fold. Strong structural similarity between two proteins is commonly interpreted as functional similarity and evolutionary relatedness.[2] Protein structures are usually more conserved than protein sequences.[3–5] Moreover, unrelated sequences can converge to the same fold.[6] As more protein structures are determined, large-scale all-against-all structure alignment projects have become useful tools for understanding the relationships between protein sequence, structure, and function.[7–9]

The second goal of structure alignment is to determine the exact similarities and differences between two proteins by delineating structurally equivalent pairs of residues. With massive sequence information becoming available, structure prediction algorithms such as homology modeling and threading are gaining increasing attention. Such techniques learn from known structures, called templates, to infer the structure of a new sequence. Structure alignment is an integral part of many structure prediction algorithms. Template protein structures are aligned to determine a common conserved core that will serve as the basis for the modeled structure. Once the structure of the new protein is solved, it is aligned to the predicted model. In this manner, structure alignment is used as a "gold standard" for testing structure prediction algorithms.

A large number of structure alignment algorithms have previously been developed.[1–31] These methods can be classified according to three characteristics. The first characteristic is the method's target function, which provides a quantitative measure of the quality of an alignment. Some methods compare the distance matrices of two 3D structures and try to minimize the differences in intramolecular distances of aligned substructures.[11,12,20] Other methods rigidly superpose one protein on the other and try to minimize the distances between paired residues.[23,25,26]

The second characteristic is the search algorithm used to find the target function's optimum value. Dynamic programming has been a popular choice because of structure alignment's apparent parallel with sequence alignment.[12,23,25,27] Other commonly used search strategies include Monte Carlo methods, or simulated annealing,[12,28] genetic algorithms,[23,26] branch-and-bound algorithms,[24] and geometric hashing.[10]

The third characteristic is whether or not the method imposes sequential constraints on the alignment. Sequential constraints substantially decrease the search space, but not without possible sacrifices. Although such constraints can enhance the algorithm's speed and precision in some cases, they may exclude the optimum alignment if the equivalent substructures in the two proteins are not

Grant sponsor: National Science Foundation; Grant number: DBI-9806002.

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

Received 9 August 1999; Accepted 8 October 1999

connected in the same order. Some search algorithms, such as dynamic programming, cannot be used without sequential constraints.

If we consider structure alignment to be a computational approach to aligning two proteins as if they were geometric objects in 3D space, the problem can be considered solved. Such an approach does not, however, address the biological relevance of the resulting alignment. This is an especially important issue when structure alignments are treated as "gold standards" for testing sequence alignment and structure prediction algorithms. This problem has not yet been resolved. Feng and Sippl[32] discovered that for many protein pairs, distinct alignments could be generated that are indistinguishable in terms of number of equivalent residues and root mean square error of superposition. Clearly, two alignments with distinct residue equivalencies cannot both be correct in the biological sense. Gerstein and Levitt[25] were also aware of the importance of this problem, and made great efforts to compare their results with manually curated alignments when testing their structure alignment algorithm. Because the target functions used by most structure alignment algorithms are only measures of the similarity between two geometric objects, target function optimization does not necessarily guarantee that the selected equivalencies correspond to residues that are conserved for biological reasons.

We decided to develop a structure alignment algorithm with the goal of generating high-quality, biologically meaningful alignments. It is understood that protein structures are more conserved in the cores than in exposed loops and turns, with the exception of those loops and turns involved in active sites. Therefore, it is our philosophy to first align the proteins' cores, as represented by their secondary structure elements (SSEs). This is achieved by minimizing the difference of distance matrices using a genetic algorithm. Once this is done, we extend the SSE alignment to include any positions in loops or turns deemed equivalent in a convergent process.

The algorithm is implemented in a computer program called KENOBI. The program has been tested on eight protein families that are highly representative, and it has proven to be robust. Specifically, KENOBI is able to generate high-quality alignments that are in complete agreement with manually curated alignments, as well as with evolutionarily conserved residue equivalencies proven by experimental results.

RESULTS

Robustness of the Algorithm

Because the genetic algorithm is heuristic and stochastic, it was important to determine whether the algorithm is able to find global optima. We aligned tryptophan synthase (1WYSA[33]), a large $(\beta\alpha)_8$ barrel with eightfold symmetry, with itself 100 times using different random number generator seeds. KENOBI found the correct alignment for all 100 seeds.

Streptavidin and avidin are two very distantly related biotin-binding proteins. Both have eight anti-parallel β-strands, which curve and twist to form a barrel with one biotin molecule bound in the center (1STP,[34] and 1AVE[35]). With sequential constraints, these proteins exhibit four-fold symmetry. Without sequential constraints, the two molecules can also be aligned upside-down. We aligned streptavidin and avidin both with and without sequential constraints. With constraints, KENOBI found the global optimum for all 100 seeds; without constraints, KENOBI found the global optimum for 84 of 100 seeds, with the remaining 16 seeds settling in local optima. The local optima include upside-down alignments, permuted alignments, and permuted upside-down alignments.

Fibronectin is a glycoprotein involved in cell adhesion. It is composed of hundreds of fibronectin type III domains, each composed of a three-stranded and a four-stranded sheet compacted together to form a compressed β-barrel. This is also a tough example, because there are many local optima that can trap structure alignment algorithms. The 3D structures of domains 7, 8, 9, and 10 of fibronectin have been solved (1FNF[36]). We arbitrarily selected domain seven as the reference structure and aligned domains 8, 9, and 10 to it (Fig. 1a). With sequential constraints, KENOBI found the global optimum for each alignment for 100 of 100 different seeds. Without the constraints, KENOBI found the global optimum for 97 to 100 of 100 different seeds.

The results of these three tests are detailed in Table I. In summary, we have an algorithm that is not sensitive to the stochastic nature of the genetic algorithm and is robust against internal repeats that can lead to a large number of suboptima.

Alignment Quality

Because one of the two main applications of a structure alignment algorithm is to create a "gold standard" for sequence alignment and structure prediction methods, we set out to test the quality of alignments generated by KENOBI. Convinced by Gerstein and Levitt[25] that it is important to compare automatic alignments against manually curated alignments, we chose to align the same three protein families as in reference 25: the all-α globins,[37] the all-β immunoglobulins,[38] and the α/β dihydrofolate reductase family.[39] In addition, we also chose the all-β protease family for which Greer[40] determined the manual alignments. For each family, one protein structure was arbitrarily selected as the reference and the other structures aligned to it one at a time using KENOBI. Five situations can arise when a KENOBI alignment is compared to a manual alignment:

1. A residue can be nonpaired in both the manual and the KENOBI alignment.
2. A residue can be nonpaired in the manual alignment but paired in the KENOBI alignment.
3. A residue can be paired in the manual alignment and nonpaired in the KENOBI alignment.
4. A residue can be paired with the same residue in the two alignments.
5. A residue can be paired with different residues in the two alignments.

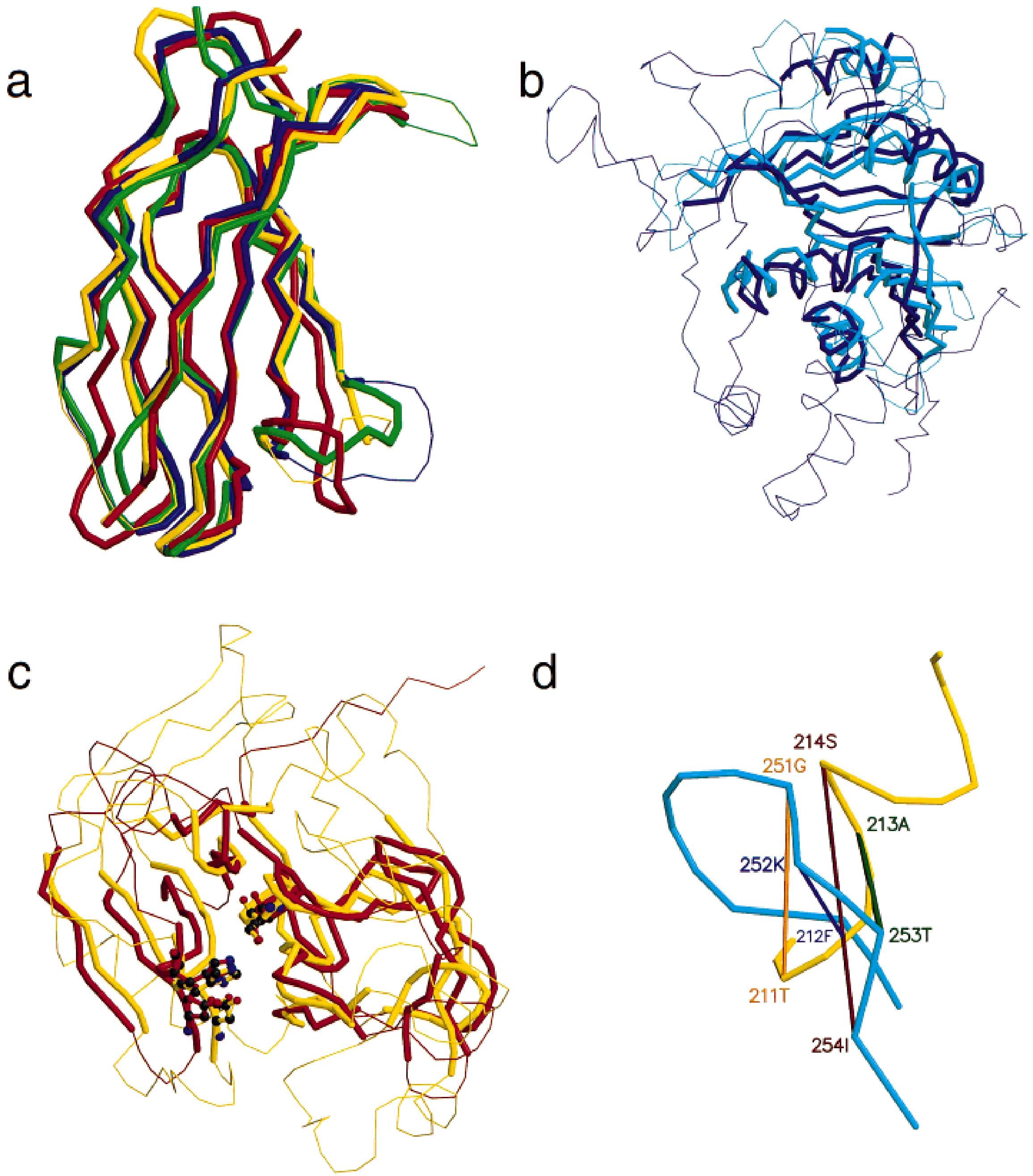

Fig. 1. Superposed 3D structures according to structure alignments. a–c: KENOBI alignments. Thick lines indicate alignable positions; thin lines indicate unalignable positions. d: A DALI alignment (see text in the Discussion section). **a:** Structure alignment of four fibronectin domains. 1FNF7 (red) served as the reference structure, and 1FNF8 (yellow), 1FNF9 (blue), and 1FNF10 (green) were aligned to it. **b:** Two aligned DNA methyltransferases, 1BOO (blue) and 2ADMA (cyan). The catalytic domains of these two proteins share very similar 3D structures, as indicated by the thick lines, despite different topological connections. **c:** Structure alignment of a human protease (1TRNA, yellow) with a viral protease (1KXF, red). The amino acids that form the Asp-His-Ser catalytic triad of each protein, shown here in ball-and-stick form, align very well. **d:** Portion of protease structures 1SGT (yellow) and 1KXF (cyan) superposed based on alignment generated by the DALI server. DALI aligned 1SGT residues 211–214 with 1KXF residues 251–254. The aligned residue pairs are labeled in the same color with a line drawn between their alpha carbons. The distances between aligned residues are: 211T–251G, 6.8 Å; 212F–252K, 3.1 Å; 213A–252T, 4.8 Å; and 214S–254I, 9.6 Å. These images were generated using MOLSCRIPT[68] and RASTER3D.[69]

Only the fifth situation is considered to be a discrepancy between the two alignments and it will be the focus of the discussion below for each of the four families.

For each KENOBI alignment, we repeated 100 runs using different random number generator seeds. The results for the four families are summarized in Table II. Generally speaking, KENOBI assigned fewer equivalent pairs than the manual methods did in the regions deemed

TABLE I. Convergence Data for Five Structure Alignments†

	Number of Correct Alignments	
Alignment	With constraints	W/o constraints
1WYSA to 1WYSA	100	100
1STP to 1AVEA	100	80
1FNF7 to 1FNF8	100	97
1FNF7 to 1FNF9	100	100
1FNF7 to 1FNF10	100	99

†To test KENOBI's ability to converge to optimal alignments, several pairs of proteins were aligned both with and without sequential constraints using 100 different random number generator seeds. These particular proteins were chosen because they contain internal repeats that can trap structure alignment algorithms in locally optimal alignments.

TABLE II. Convergence Data for Structure Alignments of Four Protein Families†

Alignment	Number of correct alignments
Globins	
2HHBA 2HHBB	100
2HHBA 5MBN	100
2HHBA 1ECD	100
2HHBA 2LHB	100
2HHBA 1LH1	90
2HHBA 2HBG	100
Immunoglobulin	
1REIA 7FABL2	85
Dihydrofolate reductases	
1DHFA 8DFR	100
1DHFA 4DFRA	100
1DHFA 3DFR	100
Serine proteases	
2CHA 5PTP	100
2CHA 1EST	100
2CHA 2KAIAB	100
2CHA 3RP2A	100
2CHA 1SGT	100
2CHA 1TON	100
Representative proteases	
1TRNA 2KAIAB	100
1TRNA 1SGT	100
1TRNA 1HAVA	100
1TRNA 1TAL	100
1TRNA 1KXF	98

†Each alignment was performed with 100 different random number generator seeds and was compared to manually curated structure alignments from the literature.

structurally equivalent by manual methods. On visual inspection of the alignments and superposed 3D structures, we discovered that the "missing" residue pairs were always excluded by the nearest neighbor and stretch-of-four constraints we impose on all residue pairs (see Methods).

Globins

For the globin family, we chose the alpha chain of human hemoglobin (2HHBA[41]) as the reference structure, and aligned six other globins to it (2HHBB, 5MBN,[42] 1ECD,[43] 2LHB,[44] 1LH1,[45] and 2HBG[46]). Five of the six globin alignments converged to the global optima for all 100 seeds, whereas the sixth converged for 90 seeds. The only discrepancy between the KENOBI alignments and the manual alignments is a helix of hemoglobin 1LH1 (PELQAHAGKVFKLVYE, positions 58–73). KENOBI aligned this helix with positions 56–71 of the representative structure 2HHBA. The manual alignment pairs them with positions 53–68 of 2HHBA, approximately one helix turn away. After reexamining our alignment, we realized that for this pair of proteins, the KENOBI alignment actually had a slightly better similarity score and a slightly smaller root mean square deviation (RMSD) than the manual alignment. We then aligned 1LH1 with each of the other globins. The only pairwise alignment that agreed with the manual alignment was 1LH1 with 2HBG; all others contained the shift described above.

Immunoglobulins

Aligning an immunoglobulin light-chain variable domain (7FABL[44]) with an immunoglobulin constant domain (1REIA[45]) is considered a difficult case by Gerstein and Levitt.[25] In fact, the basic version of their algorithm could not align this pair correctly, indicated by mismatched disulfide bonds. Only after using C^{β} coordinates could the correct alignment be achieved. We also found this alignment to be a difficult case. Only 85 of 100 runs achieved the correct alignment, whereas 14 of the remaining runs resulted in alignments that were shifted by two positions, and the last run resulted in an alignment that was shifted by one position. The shifted alignments were clearly inferior to the correct alignment, in terms of similarity score, total number of residue pairs, and RMSD after rigid fitting over the residue pairs.

Dihydrofolate reductases

All three pairwise alignments of the dihydrofolate reductase family (1DHFA,[47] 8DFR,[48] 4DFRA,[49] and 3DFR[49]) converged to the global optima for all 100 seeds. Only one discrepancy was observed between the manual and KENOBI alignments, located between residues P21W22N23[#] of 4DFRA and residues P23W24P25P26 of 1DHFA. The manual alignment assigned P21, W22, and N23 of 4DFRA to P23, W24, and P25 of 1DHFA, respectively. KENOBI aligned P21, W22, and N23 of 4DFRA with W24, P25, and P26 of 1DHFA, respectively. In terms of residue properties, the manual alignment makes sense. Because both P and W are rarely observed residues, it is very favorable to align PW with PW. After double checking the KENOBI alignment, however, we confirmed that the KENOBI alignment was actually correct. There is no ambiguity in the assignment of these two residue pairs. Superposition of the two proteins based on the manually determined residue pairs results in the same alignment generated by KENOBI in the region in question. The

[#] Residues are denoted using one letter code followed by the residue number in the original Protein Data Bank file. For example, P21W22 means proline at position 21 and tryptophan at position 22.

situation is particularly interesting if we compare the 1DHFA-3DFR structure alignment with the 1DHFA-4DFRA alignment. If we consider the regions bordered by a conserved glycine and a conserved leucine residue (G17 and L27 in 1DHFA, G15 and L24 in 4DFRA, and G14 and L23 in 3DFR), 1DHFA has a 10-residue-long loop and both 4DFRA and 3DFR have 9-residue-long loops. 1DHFA and 3DFR have similar sequences in this region, with minor differences at the end of the loop. 4DFRA's sequence at this region differs greatly from that of 1DHFA, especially at the beginning of the loop. In particular, a glycine (G20 in 1DHFA and G17 in 3DFR) is replaced by an asparagine in 4DFRA. The structure alignments correctly reflect these differences.

Proteases

The KENOBI alignments for the seven proteases (2CHA,[50] 5PTP,[51] 1EST, [52] 2KAIAB, [53] 3RP2, [54] 1SGT, [55] and 1TON[56]) completely agree with the manual alignments generated by Greer. Each alignment converged for all 100 seeds.

Difficult Cases

In the course of developing our algorithm, we needed challenging test cases. Knowing that β-strands are usually more difficult to align than α-helices, we chose to study the trypsin-like protease superfamily, which according to SCOP[57] is composed of four families. We arbitrarily picked six proteins across these four families: two from the prokaryotic protease family (1SGT and 1TAL[58]), two from the eukaryotic protease family (2KAIAB and 1TRNA[59]), one from the viral protease family (1KXF[60]), and one from the 3C cysteine protease family (1HAV[61]). All of these proteins share a common fold, with the proteolytic active site located at the crevice between two antiparallel β-barrels. Although the β-strands forming the core more or less exist in all six proteins, they are distorted, shifted, shortened, or broken in some members (especially in 1HAV and 1KXF), making these structure alignments very difficult. For these cases, there are no manually curated alignments to serve as references. Nevertheless, the KENOBI alignments agree well with published functional alignments, [62,63] with all of the active site residues perfectly aligned in all six alignments. The overall convergence of 100 runs with different random number seeds is summarized in Table II. Nearly all of the runs converged to alignments with active site positions paired correctly, as illustrated for 1KXF and 1TRNA in Figure 1c.

To test the version of the program without sequential constraints, we selected two DNA methyltransferases: M.PvuII (1BOO[64]), and M.TaqI (2ADMA[65]). M.PvuII transfers a methyl group from *S*-adenosyl-L-methionine (AdoMet) to the C5 atom of a cytosine base, while M.TaqI transfers a methyl group from AdoMet to the N4 atom of a cytosine base. Although these two proteins exhibit no sequence similarity (BLAST[66] computes an E value > 10), their catalytic domain cores have very similar 3D structures. Each of these catalytic domains is composed of a seven-stranded twisted β-sheet surrounded by six α-helices, three on each side. Seven of the eight strands point in the same direction, forming the bottom of the substrate binding cleft. Each of these catalytic domains represents a portion of a larger protein; M.PvuII has 163 residues, 8 strands, and 8 helices, and M.TaqI has 421 residues, 13 strands, and 9 helices. Moreover, the secondary structure elements in the conserved cores of these two proteins are not in the same order. As shown in Figure 2, these SSEs are permuted. This pair of proteins poses a challenge to structure alignment algorithms. The structure alignment produced by KENOBI is in complete agreement with biological findings.[64] The structural superposition of the aligned proteins is shown in Figure 1b. Of 100 random seeds, 74 produce the alignment as shown in Figure 1b, and 6 more fail to align the region M.PvuII 28-36 with M.TaqI 83–91 but are otherwise correct.

DISCUSSION

We have developed a structure alignment algorithm that is conceptually simple and requires no parameter fiddling. We have tested the algorithm on multiple criteria: convergence to the global optimum, the ability to produce biologically meaningful alignments, the performance for distantly related proteins, and the performance for permuted structures. The results above are more than satisfactory. Specifically, the algorithm has been shown to generate highly accurate, biologically relevant alignments over a broad range of protein families.

Several considerations went into the design of the algorithm to favor biologically meaningful alignments. By dividing the algorithm into two stages, we focus first on aligning the protein cores, as represented by SSEs, which tend to be more conserved than loops and turns. Treating the proteins as collections of SSEs speeds up the genetic algorithm by substantially decreasing the search space. In addition, this ensures that every SSE is treated as an intact unit. This prevents biologically incorrect alignments in which two fragments from one SSE are aligned with fragments from two different SSEs in the other protein. In the algorithm's second stage, additional residue equivalencies are added only if these residues are aligned in the context of stage 1 equivalencies. Finally, every residue pair is subject to three constraints. First, paired residues must be each other's nearest neighbors. Second, paired residues must be separated by less than some threshold distance. Third, a pair of aligned residues must exist within a stretch of four or more consecutive aligned pairs. These constraints were designed to exclude residue pairs that may be close in space but otherwise are not equivalent, e.g., if two protein backbones cross each other when superposed, a pair of residues may have exactly the same 3D coordinates and yet not be biologically equivalent.

Another important consideration was the choice of target function. We use the "elastic similarity score" developed by Holm and Sander.[20] This function uses intramolecular distance matrices to compare substructures of two proteins (see Methods). It is possible to align two proteins by searching their distance matrices for similar submatrices. Treating the proteins as collections of SSEs provides a

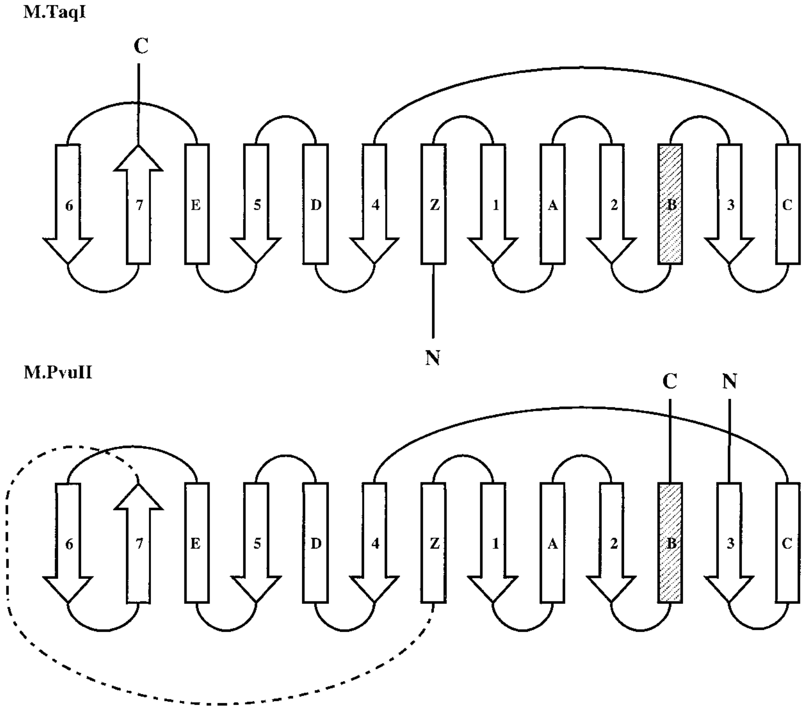

Fig. 2. Topology diagrams of two DNA methyltransferases, M.TaqI (2ADMA) and M.PvuII (1BOO). Strands are drawn as arrows, and helices as cylinders. These diagrams illustrate the circular permutation of the catalytic domains of these two proteins. For clarity, only those SSEs aligned by KENOBI are shown, with the exception of helix B (shaded), which was not aligned by KENOBI. SSE labels are those from Schluckebier et al.[70]

biologically meaningful way to search these matrices for areas of similarity.

The genetic algorithm is well suited to this application, given our representation of proteins as collections of SSEs. We found the genetic algorithm to be fast and efficient, capable of generating correct alignments from a small number of randomly generated SSE alignments (100 by default). The genetic algorithm is capable of both coarse and detailed manipulations of the alignments. The genetic algorithm's speed is in large part a result of the powerful recombination operator, which allows for large-scale exchanges of information between pairs of alignments (see Methods).

There exist many structure alignment tools and structure classification databases. SCOP,[7] CATH,[51] and MMDB (built with the VAST algorithm[30,31]) are the leading protein classification databases. They categorize proteins into families according to structures and/or functions. Of these, only MMDB provides detailed structure alignments. DALI is a well-known structure alignment program available as a web-based server. Users can query a structure against a protein structure database or align two specific structures. Although VAST and DALI focus on detecting remote structural similarities, KENOBI's aim is to generate high-quality, biologically meaningful residue equivalencies. The difference in these goals is apparent when comparing KENOBI and DALI alignments of the same protein pairs. Alignments generated by DALI typically include more aligned residues and subsequently have higher RMSDs than KENOBI alignments.

Two cases in particular illustrate the differences between KENOBI and DALI. The first of these is the alignment of the two DNA methyltransferases (1BOO and 2ADMA). The DALI alignment, generated by the DALI e-mail server, correctly aligns strands 3–7 and helices C–E but does not align the permuted portions of the structures, namely, strands 1 and 2 and helices A and Z (Fig. 2). This is to be expected because the DALI server performs only sequentially constrained alignments.

The second case involves the alignment of two distantly related serine proteases, 1SGT and 1KXF. Although the DALI alignment correctly matches the Asp-His-Ser catalytic triads, it also includes several questionable equivalencies. For example, DALI aligns 1SGT residues 211–214 with 1KXF residues 251–254. When the structures are superposed according to the DALI alignment, close inspection of this region reveals that a helical region in 1SGT (residues 221–214) has been aligned with an extended region in 1KXF (residues 251–254) and the paired residues are in fact quite distant from each other (Fig. 1d). These extraneous equivalencies and others like them are excluded by the strict nearest neighbor, stretch-of-four, and distance constraints built into KENOBI. It should be noted that MMDB does not list the protein pairs from either of these cases as structural neighbors; therefore, it is not possible to compare KENOBI and VAST/MMDB for these cases.

The current version of KENOBI faces several limitations. Because of the first stage SSE alignment, KENOBI is unable to align small proteins or peptides that do not have any secondary structures. KENOBI currently recognizes two types of secondary structure elements: α-helices and β-strands as calculated by DSSP. We are in the process of adding a third type of SSE that can be assigned by the user. The user-defined SSEs could be applied to conserved tight turns or other nonsecondary-structure regions such as cysteine-rich motifs. KENOBI's convergence, although generally quite good (85% or more) is not perfect and could be improved with better initial populations and smarter operators. For example, we could take advantage of the graph isomorphism approach by Grindley and colleagues[19] and the "3D-lookup" approach by Holm and Sander[24] to rapidly find seed SSE alignments. Finally, although KENOBI is sufficiently fast for moderate numbers of pairwise alignments, it is not fast enough to allow for the 10,000 or so comparisons needed to query a protein structure against the Protein Data Bank. We hope that by parallelizing the KENOBI code, we can improve its performance such that it could be used for efficient database searching.

METHODS

Stage 1: Target Function

It is critical that stage 1 of the algorithm results in the correct pairing of SSEs. We have adopted the "elastic similarity score" developed by Holm and Sander,[20] which is based on intramolecular distances.* It is defined as (1)

$$S = \left\{ \begin{array}{ll} \sum_{i=1}^{L} \sum_{j=1}^{L} \left(\theta - \frac{|d_{ij}^{A} - d_{ij}^{B}|}{d_{ij}^{*}} \right) e^{-(d_{ij}^{*}/\alpha)^2}, & i \neq j \\ \theta, & i = j \end{array} \right\},$$

where i and j label pairs of matched residues (e.g., $i(i_A, i_B)$ is the ith residue pair, with the $i_{\mathrm{A}}^{\mathrm{th}}$ residue in *protein A* paired with the $i_{\mathrm{B}}^{\mathrm{th}}$ residue in *protein B*); L is the total number of residue pairs; d_{ij}^{A} is an element of the distance matrix of protein A, denoting the distance between the i_{A}^{th} residue and the j_{A}^{th} residue in protein A; likewise d_{ij}^{B} denotes the distance between the i_{B}^{th} residue and the j_{B}^{th} residue in protein B; d_{ij}^{*} is the average of d_{ij}^{A} and $d_{ij}{}^{B}$; θ is a constant similarity threshold (set to 0.20); and $e^{-(d_{ij}^{*}/\alpha)^2}$ is an envelope function (α = 20 Å) designed to reduce the contribution of distant pairs to the overall score. Residues that are not aligned do not contribute to the similarity score S.

Equation 1 helps balance structure alignment's two main objectives: to simultaneously maximize the number of equivalent residue pairs and minimize the distance between these pairs. Equation 1 can be easily explained using a simple example. Assume residues 1, 2, and 3 of protein A are aligned with residues 1, 2, and 3 of protein B, respectively (L = 3 in Equation 1). If they align well, we would expect d_{12}^{A} to be similar to d_{12}^{B}, d_{13}^{A} to be similar to d_{13}^{B}, and d_{23}^{A} to be similar to d_{B}^{23}. Then the similarity score S would be slightly smaller than 9θ.

Stage 1: Secondary Structure Elements (SSEs)

Similarities in the structures of two proteins can be detected by similarities in their distance matrices. In this sense, the structure alignment problem becomes a search for regions of similarity shared by the two distance matrices. This search space is so large that current computing power cannot exhaustively enumerate all possible structure alignments between two proteins longer than 25 residues. By treating each protein as a collection of SSEs, two protein structures can be aligned by searching for the "best" set of SSE pairings, which we call SSE alignments. This also dramatically reduces the search space. If three α-helices from *protein A* were similar to three α-helices from *protein B*, the distance matrices of these two proteins would contain three similar submatrices along the main diagonals, one for each helix. In addition, there would appear in each distance matrix a similar off-diagonal submatrix corresponding to the contact patterns between the three helices (Fig. 3). Allowing SSEs to align only with other SSEs of the same type further reduces the search space.

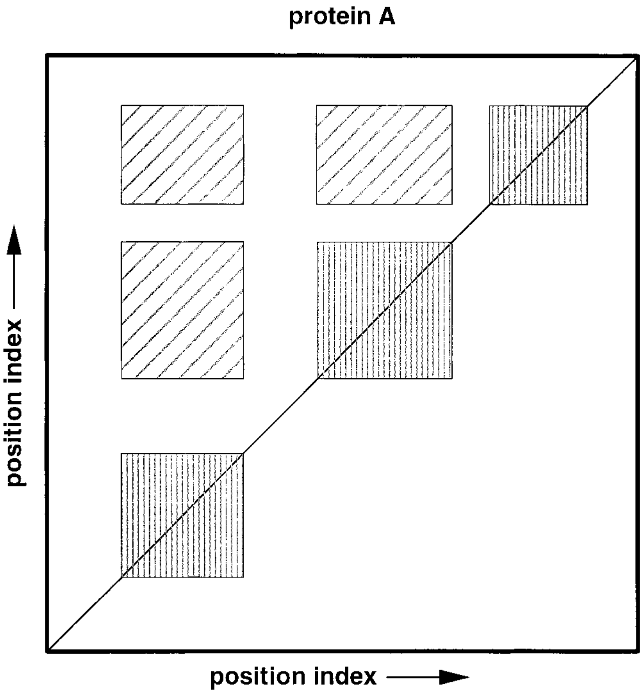

Fig. 3. An example intramolecular distance matrix. The three submatrices along the main diagonal (vertically lined boxes) represent three SSEs. Each submatrix contains the intramolecular distances for the residues in one SSE. The three off-diagonal submatrices (diagonally lined boxes) represent the contact patterns between the three SSEs.

Protein structures are provided to the program in the form of Protein Data Bank files. SSEs are first calculated using DSSP,[67] and then subjected to a smoothing algorithm (http://bmerc-www.bu.edu/needle-doc/new/dssp-progs.html). Residues that are neither helix ("H") nor strand ("E") are converted to loops ("L"). Secondary structure elements with fewer than four residues are also converted into loops. The beginning and ending residue numbers of an SSE define fixed boundaries that can neither be changed nor traversed by the genetic algorithm.

*Because the C^{α} atom is at the center of the residue and its coordinates are almost always adequate for representing the residue's position, our algorithm uses only C^{α} atoms and ignores all other atoms.

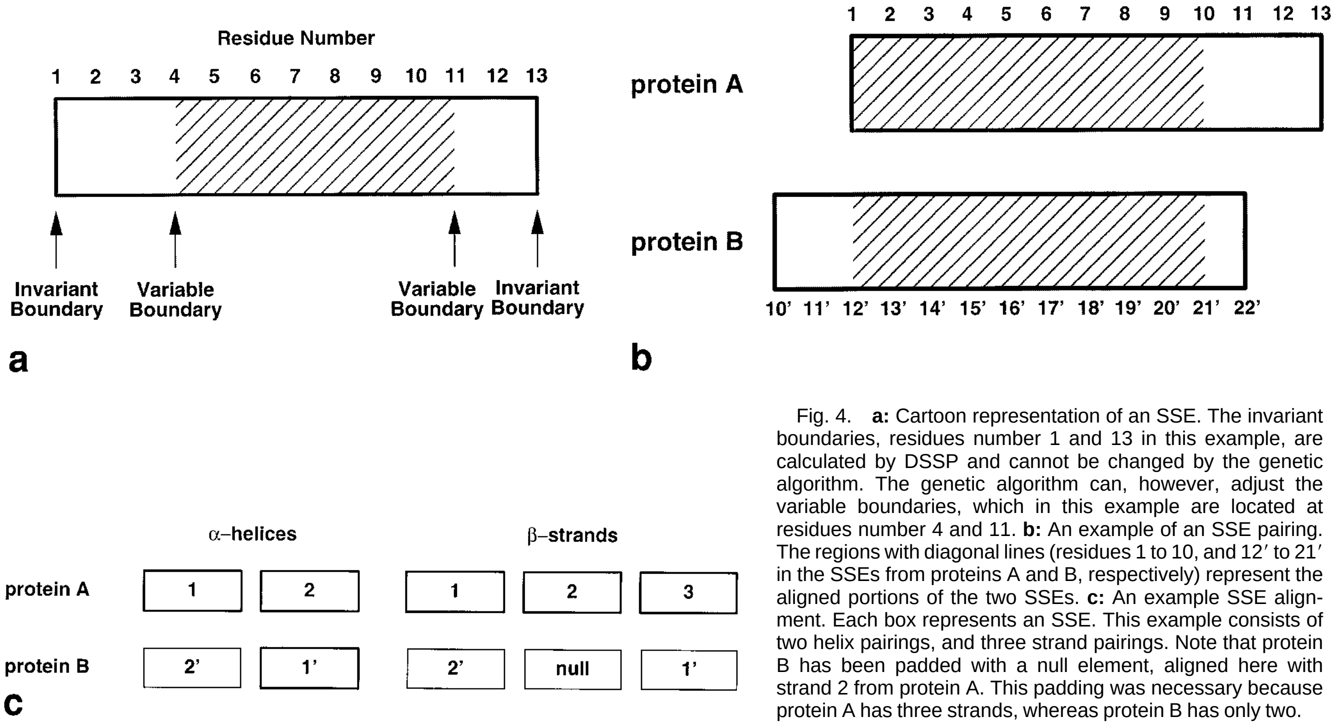

Fig. 4. **a:** Cartoon representation of an SSE. The invariant boundaries, residues number 1 and 13 in this example, are calculated by DSSP and cannot be changed by the genetic algorithm. The genetic algorithm can, however, adjust the variable boundaries, which in this example are located at residues number 4 and 11. **b:** An example of an SSE pairing. The regions with diagonal lines (residues 1 to 10, and 12′ to 21′ in the SSEs from proteins A and B, respectively) represent the aligned portions of the two SSEs. **c:** An example SSE alignment. Each box represents an SSE. This example consists of two helix pairings, and three strand pairings. Note that protein B has been padded with a null element, aligned here with strand 2 from protein A. This padding was necessary because protein A has three strands, whereas protein B has only two.

Stage 1: The Genetic Algorithm

It is computationally prohibitive to search all possible SSE alignments, including the detailed residue pairings, exhaustively. Instead, we use a genetic algorithm to search for the optimal solution to Equation 1 heuristically. The genetic algorithm's basic scheme is as follows:

1. Generate an initial population of possible SSE alignments.
2. Alter each alignment using the "mutate," "hop," and "swap" operators.
3. Carry out "recombination" between randomly assigned pairs of alignments using the "crossover" operator.
4. Accept or reject the alterations made to each alignment.
5. Exit if certain conditions are met. Otherwise go to step 2.

Initial population

Each SSE alignment is represented as lists of paired SSEs (Fig. 4). Every SSE is paired with an SSE of the same type from the other protein. If the two proteins have an unequal number of SSEs of a given type (e.g., *protein A* has four helices, and *protein B* has two helices), the protein with fewer SSEs is padded with null elements. An SSE paired with a null element is in effect not aligned and does not contribute to the alignment's similarity score, as defined by Equation1. We require an SSE pair to have four or more residue pairs. The initial length of each SSE pair is constrained by the secondary structure elements' fixed boundaries and is determined randomly. If two SSEs of unequal length are paired, the genetic algorithm selects an equal number of residues from each SSE.

Because the SSE alignment search space is very large (see Discussion), we bias the initial population toward SSE pair doublets (two SSEs from one protein paired with two SSEs from the other protein) that score favorably according to Equation 1. The similarity scores of all SSE pair doublets are first calculated. Initial alignments are then generated by selecting SSE pair doublets from this list until all SSEs have been chosen. Doublets are selected with probability (2)

$$P_k = \frac{e^{S_k}}{\sum_k e^{S_k}},$$

where S_k is the score of the kth doublet as calculated using Equation 1. This prescreening for high-scoring doublets biases the initial alignments toward better guesses. The number of alignments (population size) is constant and set to 100 by default. Before we start to alter an alignment, we record all key information necessary to return it to its unaltered state.

The "mutate" operator

Mutation provides a method to fine-tune the individual SSE pairs. All mutations are constrained by the fixed boundaries of the two SSEs and the requirement that every SSE pair must contain at least four residue pairs. Mutations can increase the number of residue pairs by one, decrease the number of residue pairs by one, or shift one SSE relative to the other by one position. The exact nature of a mutation is determined randomly and accomplished by changing the working boundaries of the paired

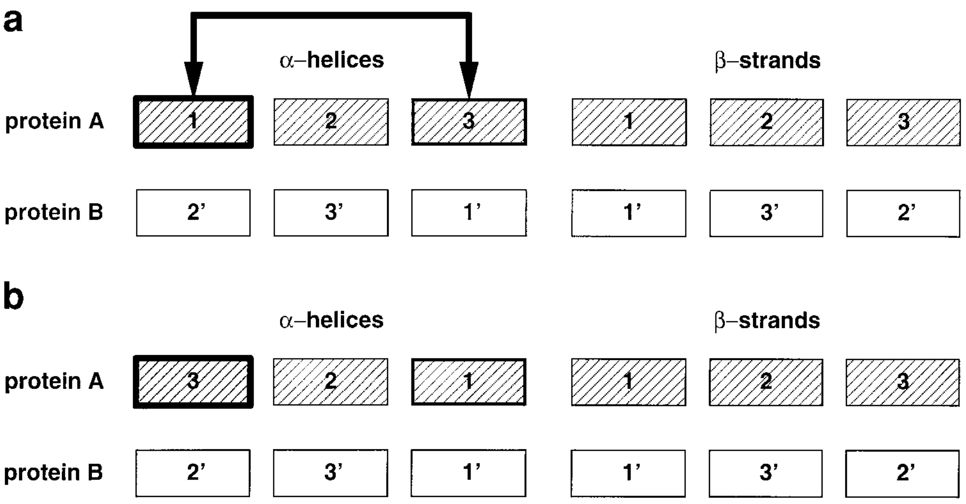

Fig. 5. The "hop" operator. **a:** First, two SSEs of the same type from one protein are selected (helices 1 and 3 from protein A). **b:** These two SSEs then exchange places in the alignment. The resulting new SSE pairs are then trimmed if necessary (not shown).

SSEs. The default mutation probability for every SSE pair is 3%.

The "hop" operator

SSE pairs are allowed to "hop" along the alignment. If an alignment is selected for hopping, two SSE pairs of the same type from one protein are selected. The first SSE pair is chosen with probability determined using Equation 2. The second is chosen blindly from the remaining SSE pairs of the same type as the first. Once chosen, the two SSEs trade places (Fig. 5). When necessary, the resulting new pairs are trimmed to give the paired SSEs equal lengths. The default hop probability for each alignment is 5%.

The "swap" operator

Each alignment is subject to the swap operator with equal probability (by default 5%). When an alignment is to be swapped, it is randomly assigned a partner from the rest of the population. A swap between two alignments consists of a wholesale trade of all of the SSE pairs of one type in one alignment for those of the same type in the other alignment. For example, if *alignments X* and *Y* were chosen to swap with each other, *alignment X* would trade all of its β-strand pairs for all of *alignment Y*'s β-strand pairs (Fig. 6). This exchange is most productive for cases in which one alignment has well aligned α-helices and the other has well aligned β-strands

The "crossover" operator

Every alignment is randomly assigned a crossover partner from the rest of the population. The crossover operator then carries out several actions. First, the SSE pairs in the α-helix and β-strand lists are sorted. Next, a crossover point is randomly assigned for each list. The two alignments then exchange all SSE pairs on one side of the crossover points (Fig. 7). The helix and strand lists are then repaired to ensure that they contain exactly one copy of each SSE from each protein.

Reevaluation of the population

Each altered alignment is evaluated according to Equation 1. The contribution of each SSE pair to the total similarity score is also calculated. The alignment's total score is compared to its score before the alteration. If the alignment's current score is greater than its previous score, all changes made to the alignment are accepted. If the alignment's current score is less than its previous score, the alignment is returned to its previous condition using the recorded information. After each round, the program calculates the average score for the population. The program also maintains a list of the 10 best alignments seen and the best score seen at any point in the run.

Exit the Genetic Algorithm

The genetic algorithm exits if it reaches a preset number of rounds, or the best score remains unchanged for 20 consecutive rounds, or the average score for the population is equal to the best score seen.

Stage 1: Refine the Best Alignment Generated by the Genetic Algorithm

After completion of the genetic algorithm, the alignment with the best score is refined. The SSE pairs that make a negative contribution to the overall score are eliminated from the alignment. The remaining pairs are subjected to a series of small alterations in an attempt to improve the score of the overall alignment. First, the SSEs in each pair are shifted incrementally up to four positions in each direction. Once the optimum shift has been determined, SSEs are extended in each direction in an attempt to further improve the alignment's score.

In some cases, the best alignment generated by the genetic algorithm and refinement procedure has correct SSE pairings, but incorrect residue alignments within some SSE pairs. Such incorrect pairings are usually shifted by two positions if they are strands or three positions if they are helices. We have designed a "shake" operator to tackle this problem. The "shake" operator randomly shifts the relative position of every SSE pairing by zero to three positions in either direction. The alignment is then reevaluated. If the alignment's new score is better than its previous score, the adjustments are accepted. If the alignment's current score is worse than its

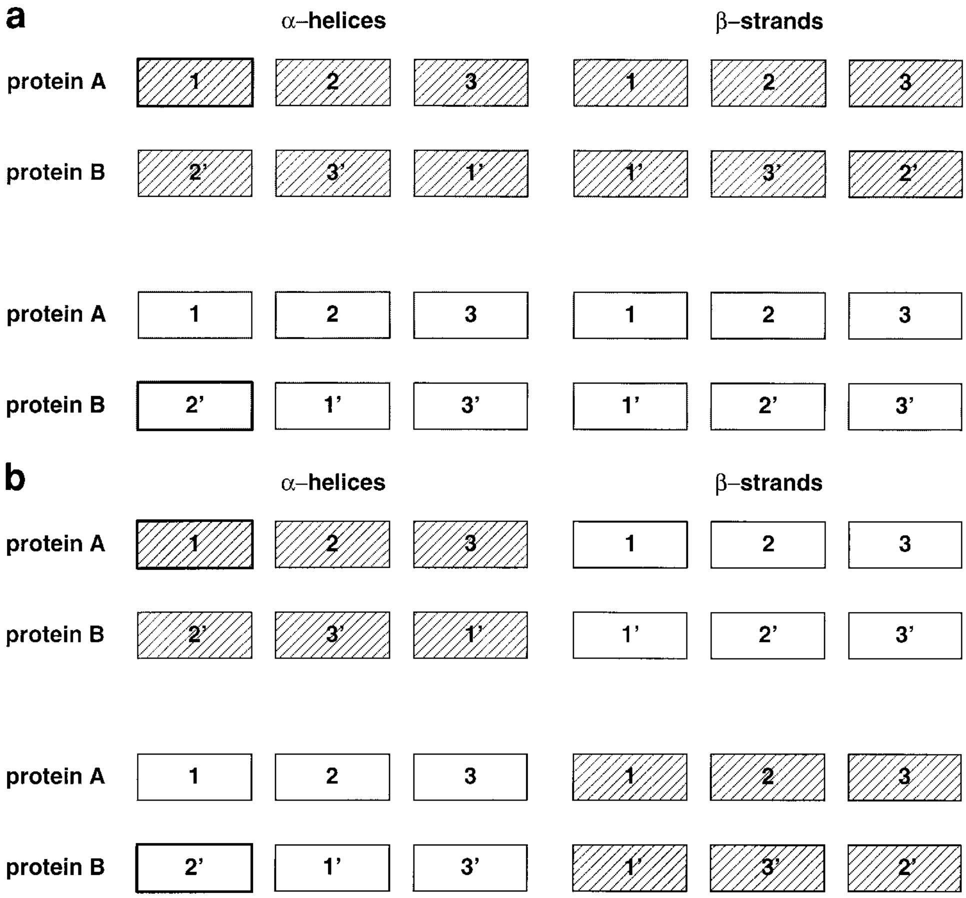

Fig. 6. The "swap" operator. **a:** Two SSE alignments are selected to swap. SSEs initially in the upper alignment appear with diagonal lines. **b:** In this example, the strand pairs from the upper alignment are swapped with the strand pairs from the lower alignment.

previous score, the alignment is returned to its previous condition. This process is repeated a set number of times (5,000 by default). The "shake" operator is computationally expensive and can be turned off for less difficult alignments.

Stage 2: Extend the Alignment to Nonsecondary Structure Regions

Using the residue pairings generated at stage 1, the two protein structures are rigidly superposed using a least-squares fit algorithm to minimize the RMSD between the aligned residue pairs. We then search the protein backbones for additional equivalent residue pairs, especially in non-SSE regions. We recruit a residue pair if the paired residues are each other's nearest neighbors and are separated by a distance less than some threshold. Any equivalent pair that does not occur in a stretch of at least four consecutive pairs is thrown out to eliminate spurious equivalencies. The combination of nearest-neighbor and stretch-of-four criteria turns out to be surprisingly strict. The distance threshold is set to a large value (by default 10 Å) to exclude only those pairs that are obviously not equivalent. The nearest neighbor and stretch-of-four criteria exclude other incorrect pairs. This cycle of RMS fitting, recruitment, and pruning continues until the alignment converges.

Computer Implementation

The algorithm has been implemented in a C++ program called KENOBI. Sequential constraints can be turned on or off from the command line. The hop and crossover operators each have sequentially constrained and nonconstrained modes. The constrained mode functions as described above with the added requirement that all operations maintain sequential ordering of the SSEs. All parameters can also be adjusted from the command line. To test KENOBI's speed, we performed an easy alignment of two globins, 2HHBA (141 residues, 6 helices) and 2HHBB (146 residues, 6 helices) and a difficult alignment of two DNA methyltransferases, 1BOO (163 residues, 8 strands, and 8 helices) and 2ADMA (421 residues, 13 strands, and 9 helices) on several different computer systems. The results, summarized in Table III, demonstrate that KENOBI is quite fast. KENOBI aligned the globins in 47 seconds without sequential constraints on a personal computer. With sequential constraints, KENOBI required only 21 seconds to align the globins. Even the large methyltransferases required only a few minutes to align.

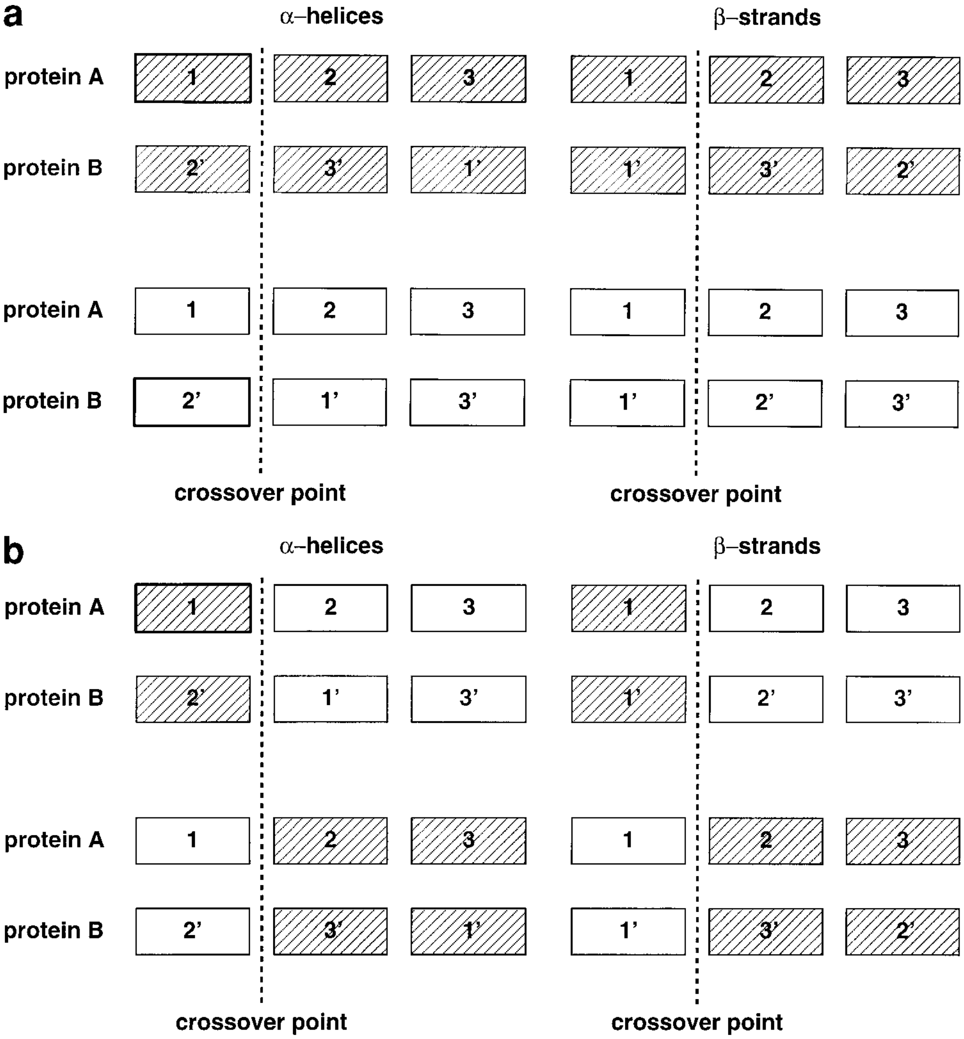

Fig. 7. The "crossover" operator. **a:** First, two SSE alignments are selected to crossover. SSEs initially in the upper alignment appear with diagonal lines. Crossover points, represented here as dashed vertical lines, are then assigned to both the helix and strand pairings. **b:** The two alignments then exchange all SSE pairings to the right of the crossover point. In this example, the new alignments contain exactly one copy of each SSE, so the alignments do not need to be repaired.

TABLE III. Speed of Alignments by KENOBI on Several Different Computer Architectures[†]

			CPU time (sec)		
System	Processor type	Number of processors	1BOO and 2ADMA without constraints	2HHBA and 2HHBB without constraints	2HHBA and 2HHBB with constraints
SGI Indigo 2	R8000, 75 MHz	1	407	122	73
SGI O2	R5000, 200 MHz	1	239	76	38
SGI Power Onyx	R10000, 194 MHz	1	222	55	26
SGI/CRAY Origin 2000	R10000, 195 MHz	1	175	52	25
AMD (LINUX operating system)	AMD K6-2, 400 MHz	1	178	47	21

[†] To test KENOBI's speed, we aligned a pair of DNA methyltransferases, 1BOO and 2ADMA, and a pair of globins, 1/2HHBA and 2HHBB, on several different computer systems. The times reported are the average CPU times needed to generate four correct alignments. The same random number generator seeds were used on all systems. The DNA methyltransferase alignments were made without sequential constraints; the globin alignments were made both with and without sequential constraints.

ACKNOWLEDGMENTS

We thank Dr. Charles DeLisi for his valuable advice and continuing support. We also thank Dr. Joel Janin for his insightful comments on the manuscript.

REFERENCES

1. Perutz MF, et al. Structure of Naemoglobin: a three-dimensional Fourier synthesis at 5.5 Å resolution, obtained by x-ray analysis. Nature 1960;185:416–422.
2. Rozwarski DA, et al. Structural comparisons among the short-chain helical cytokines. Structure 1994;2:159–173.
3. Chothia C, Lesk AM. The relation between the divergence of sequence and structure in proteins. EMBO J 1986;5:823–826.
4. Murzin AG. Structural classification of proteins: new superfamilies. Curr Opin Struct Biol 1996;6:386–394.
5. Murzin AG. How far divergent evolution goes in proteins. Curr Opin Struct Biol 1998; 8:380–387.
6. Orengo CA, et al. Structural similarity between the pleckstrin homology domain and verotoxin: the problem of measuring and evaluating structural similarity. Protein Sci 1995;4:1977–1983.
7. Hubbard TJ, et al. SCOP: a Structural Classification of Proteins database. Nucleic Acids Res 1999;27:254–256.
8. Orengo CA, et al. The CATH Database provides insights into protein structure/function relationships. Nucleic Acids Res 1999; 27:275–279.
9. Holm L, Sander C. Protein folds and families: sequence and structure alignments. Nucleic Acids Res 1999;27:244–247.
10. Artymiuk PJ, et al. Searching techniques for databases of protein structures. J Inform Sci 1989; 15:287–298.
11. Taylor WR, Orengo CA. Protein structure alignment. J Mol Biol 1989;208:1–22.
12. Sali A, Blundell TL. Definition of general topological equivalence in protein structures. A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. J Mol Biol 1990;212:403–428.
13. Barakat MT, Dean PM. Molecular structure matching by simulated annealing. III. The incorporation of null correspondences into the matching problem. J Comput Aided Mol Des 1991;5:107–117.
14. Vriend G, Sander C. Detection of common three-dimensional substructures in proteins. Proteins 1991;11:52–58.
15. Alexandrov NN, Takahashi K, Go N. Common spatial arrangements of backbone fragments in homologous and non-homologous proteins. J Mol Biol 1992;225:5–9.
16. Fischer D, et al. An efficient automated computer vision based technique for detection of three dimensional structural motifs in proteins. J Biomol Struct Dyn 1992;9:769–789.
17. Orengo CA, Brown NP, Taylor WR. Fast structure alignment for protein databank searching. Proteins 1992;14:139–167.
18. Russell RB, Barton GJ. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. Proteins 1992;14:309–323.
19. Grindley HM, et al. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. J Mol Biol 1993;229:707–721.
20. Holm L, Sander C. Protein structure comparison by alignment of distance matrices. J Mol Biol 1993;233:123–138.
21. Johnson M, et al. In: 27th Hawaii International Conference on Systems Sciences, Hawaii, 1994. Los Alamitos, CA: IEEE Computer Society Press; 1994.
22. Diederichs K. Structural superposition of proteins with unknown alignment and detection of topological similarity using a six-dimensional search algorithm. Proteins 1995;23:187–195.
23. May AC, Johnson MS. Improved genetic algorithm-based protein structure comparisons: pairwise and multiple superpositions. Protein Eng 1995;8:873–882.
24. Holm L, Sander C. Mapping the protein universe. Science 1996;273: 595–603.
25. 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.
26. Lehtonen JV, et al. Finding local structural similarities among families of unrelated protein structures: a generic non-linear alignment algorithm. Proteins 1999;34:341–355.
27. Cohen G. ALIGN: a program to superimpose protein coordinates, accounting for insertions and deletions. J Appl Cryst 1997;30:1160–1161.
28. Holm L, Sander C. Fast and simple Monte Carlo algorithm for side chain optimization in proteins: application to model building by homology. Proteins 1992;14:213–223.
29. Shindyalov IN, Bourne PE. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng 1998;11:739–747.
30. Madej T, Gibrat JF, Bryant SH. Threading a database of protein cores. Proteins 1995;23:356—69.
31. Gibrat JF, Madej T, Bryant SH. Surprising similarities in structure comparison. Curr Opin Struct Biol 1996;6:377–385.
32. Feng ZK, Sippl MJ. Optimum superimposition of protein structures: ambiguities and implications. Folding Design 1996;1:123–132.
33. Hyde CC, et al. Three-dimensional structure of the tryptophan synthase alpha 2 beta 2 multienzyme complex from Salmonella typhimurium. J Biol Chem 1988;263:17857–17871.
34. Weber PC, et al. Structural origins of high-affinity biotin binding to streptavidin. Science 1989; 243:85–88.
35. Pugliese L, et al. Three-dimensional structure of the tetragonal crystal form of egg-white avidin in its functional complex with biotin at 2.7 Å resolution. J Mol Biol 1993;231:698–710.
36. Leahy DJ, Aukhil I, Erickson HP. 2.0 A crystal structure of a four-domain segment of human fibronectin encompassing the RGD loop and synergy region. Cell 1996;84:155–164.
37. Bashford D, Chothia C, Lesk AM. Determinants of a protein fold. Unique features of the globin amino acid sequences. J Mol Biol 1987;196:199–216.
38. Lesk AM, Chothia C. Evolution of proteins formed by beta-sheets. II. The core of the immunoglobulin domains. J Mol Biol 1982;160: 325–342.
39. Gerstein M, Sonnhammer EL, Chothia C. Volume changes in protein evolution. J Mol Biol 1994;236:1067–1078.
40. Greer J. Comparative modeling methods: application to the family of mammalian serine proteases. Proteins 1990;7:317–334.
41. Fermi G, et al. The crystal structure of human deoxyhaemoglobin at 1.74 Å resolution. J Mol Biol 1984;175:159–174.
42. Takano T. Refinement of myoglobin and cytochrome C. In Hall SP, Ashida T, editors. Methods and applications in crystallographic computing. Oxford, England: Oxford University Press; 1984. p 262.
43. Steigemann W, Weber E. Structure of erythrocruorin in different ligand states refined at 1.4 Å resolution. J Mol Biol 1979;127:309–338.
44. Honzatko RB, Hendrickson WA, Love WE. Refinement of a molecular model for lamprey hemoglobin from Petromyzon marinus. J Mol Biol 1985;184:147–164.
45. Arutyunyan EG, et al. X-ray structural investigation of leghemoglobin. Vi. Structure of acetate- ferrileghemoglobin at a resolution of 2.0 Angstroms [Russian]. Kristallografiya 1980;25:80.
46. Arents G, Love WE. Glycera dibranchiata hemoglobin. Structure and refinement at 1.5 Å resolution [published erratum appears in J Mol Biol 1990;215:473]. J Mol Biol 1989;210:149–161.
47. Davies JF II, et al. Crystal structures of recombinant human dihydrofolate reductase complexed with folate and 5-deazafolate. Biochemistry 1990;29:9467–9479.
48. McTigue MA, et al. Crystal structure of chicken liver dihydrofolate reductase complexed with NADP+ and biopterin. Biochemistry 1992;31:7264–7273.
49. Bolin JT, et al. Crystal structures of Escherichia coli and Lactobacillus casei dihydrofolate reductase refined at 1.7 Å resolution. I. General features and binding of methotrexate. J Biol Chem 1982;257:13650–13662.
50. Birktoft JJ, Blow DM. Structure of crystalline-α-chymotrypsin. V. The atomic structure of tosyl-α-chymotrypsin at 2 Å resolution. J Mol Biol 1972;68:187–240.
51. Finer-Moore JS, et al. Solvent structure in crystals of trypsin determined by X-ray and neutron diffraction. Proteins 1992;12: 203–222.
52. Sawyer L, et al. The atomic structure of crystalline porcine pancreatic elastase at 2.5 Å resolution: comparisons with the structure of alpha-chymotrypsin. J Mol Biol 1978;118:137–208.
53. Chen Z, Bode W. Refined 2.5 Å X-ray crystal structure of the complex formed by porcine kallikrein A and the bovine pancreatic trypsin inhibitor. Crystallization, Patterson search, structure determination, refinement, structure and comparison with its

components and with the bovine trypsin–pancreatic trypsin inhibitor complex. J Mol Biol 1983;164:283–311.

54. Remington SJ, et al. The structure of rat mast cell protease II at 1.9-Å resolution. Biochemistry 1988;27:8097–8105.

55. Read RJ, James MN. Refined crystal structure of Streptomyces griseus trypsin at 1.7 Å resolution. J Mol Biol 1988;200:523–551.

56. Fujinaga M, James MN. Rat submaxillary gland serine protease, tonin. Structure solution and refinement at 1.8 Å resolution. J Mol Biol 1987;195:373–396.

57. Murzin AG, et al. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 1995;247:536–540.

58. Rader SD, Agard DA. Conformational substates in enzyme mechanism: the 120 K structure of alpha-lytic protease at 1.5 Å resolution. Protein Sci 1997;6:1375–1386.

59. Gaboriaud C, et al. Crystal structure of human trypsin 1: unexpected phosphorylation of Tyr151. J Mol Biol 1996;259:995–1010.

60. Choi HK, et al. Structural analysis of Sindbis virus capsid mutants involving assembly and catalysis [published erratum appears in J Mol Biol 1997;266:633–634]. J Mol Biol 1996;262:151–167.

61. Bergmann EM, et al. The refined crystal structure of the 3C gene product from hepatitis A virus: specific proteinase activity and RNA recognition. J Virol 1997;71:2436–2448.

62. Choi HK, et al. Structure of Sindbis virus core protein reveals a chymotrypsin-like serine proteinase and the organization of the virion [see comments]. Nature 1991;354:37–43.

63. Allaire M, et al. Picornaviral 3C cysteine proteinases have a fold similar to chymotrypsin-like serine proteinases. Nature 1994;369:72–76.

64. Gong W, et al. Structure of pvu II DNA-(cytosine N4) methyltransferase, an example of domain permutation and protein fold assignment. Nucleic Acids Res 1997;25:2702–2715.

65. Schluckebier G, et al. Differential binding of S-adenosylmethionine S-adenosylhomocysteine and Sinefungin to the adenine-specific DNA methyltransferase M.TaqI. J Mol Biol 1997;265:56–67.

66. Tatusova TA, Madden TL. BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences [In Process Citation]. FEMS Microbiol Lett 1999;174:247–250.

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

68. Kraulis PJ. MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures. J Appl Cryst 1991;24:946–950.

69. Merritt EA, Bacon DJ. Raster3D—photorealistic molecular graphics. Methods Enzymol 1997;277:505–524.

70. Schluckebier G, et al. Universal catalytic domain structure of AdoMet-dependent methyltransferases. J Mol Biol 1995;247:16–20.