Hierarchical Protein Structure Superposition using both Secondary Structure and Atomic Representations

Amit P. Singh and Douglas L. Brutlag
Section on Medical Informatics and Department of Biochemistry
Stanford University, Stanford, CA 94305
apsingh@cmgm.stanford.edu, brutlag@stanford.edu
Tel: (415) 723-4025, (415) 723-6593
Fax: (415) 725-6044

Abstract

The structural comparison of proteins has become increasingly important as a means to identify protein motifs and fold families. In this paper we present a new algorithm for the comparison of proteins based on a hierarchy of structural representations, from the secondary structure level to the atomic level. Our technique represents α-helices and β-strands as vectors and uses a set of seven scoring functions to compare pairs of vectors from different proteins. The scores obtained are used in a dynamic programming algorithm that finds the best local alignment of the two sets of vectors. The second step in our algorithm is based on the atomic coordinates of the protein structures and improves the initial vector alignment by iteratively minimizing the RMSD between pairs of nearest atoms from the two proteins. We refine the final alignment by determining a core of well aligned atoms and minimizing the RMSD of this core. In a comparison of our method to Holm and Sander's DALI algorithm, our program was able to detect structural similarity at the same level as DALI. We also performed searches of a representative set of the Protein Data Bank (PDB) using our program and detected structurally similarity between several distantly related proteins.

Introduction

The number of protein structures in the Brookhaven Protein Data Bank (Bernstein et al., 1977) has been growing rapidly and is currently (as of January, 1997) more than 5,400. The number of known fold families into which these structures can be classified, are, on the other hand, relatively few (Chothia, 1992; Holm and Sander, 1996a; Orengo et al 1993). With the growing number of known unique protein structures, it has become increasingly important to study the levels of structural similarity that exist among these proteins as a means to identify structural motifs and fold families. The comparison of proteins at a structural level is a fundamental step towards the understanding of the folding techniques that are used by biological organisms to construct stable and functional proteins. Structure is also widely believed to be closer to function than sequence, which further emphasizes the importance of studying the three-dimensional relationships within and between proteins. In addition, since structure is more highly conserved than sequence, the comparison of protein structures is essential to obtain more accurate estimates of evolutionary distances between proteins and protein families.

There have been several methods proposed to compare protein structures and measure the degree of structural similarity between them. These methods have been based on comparing scalar distance plots (Holm and Sander, 1993), computing differences of vector distance plots (Orengo and Taylor, 1996), minimizing the soap-bubble surface area between two protein backbones (Falicov and Cohen, 1996), and applying dynamic programming on pair-wise distances between proteins (Subbiah et al, 1993; Gerstein and Levitt, 1996). Several other techniques have also been reported (Holm and Sander, 1996a; Holm and Sander, 1995; Russel and Barton, 1992; Godzik and Skolnick, 1994; Zuker and Somorjai, 1989; Sali and Blundel, 1990; Taylor and Orengo, 1989; Mitchel et al, 1989; Vriend and Sander, 1991; Barakat and Dean, 1991). In this paper we describe a new algorithm for structural superposition based on a hierarchical decomposition of the protein structures from the secondary structure level to the atomic level. We represent the secondary structure elements of the proteins as vectors and obtain an initial superposition by computing a local alignment, using dynamic programming, of these secondary structure vectors. We then compute an atomic superposition, using the 3-D coordinates of the backbone atoms, by performing a greedy search that tries to minimize the root mean square deviation (RMSD) between pairs of nearest atoms from the two proteins. Our method is both robust and fast, and is capable of detecting global similarity between entire proteins as well as similarity of local structural domains. Due to the hierarchical nature of our algorithm, it is able to detect minor local structural similarities between proteins and also manage such difficult features as variable length loops or the insertion or deletion of entire secondary structure elements. We have tested our algorithm by searching the PDB using various query structures and have

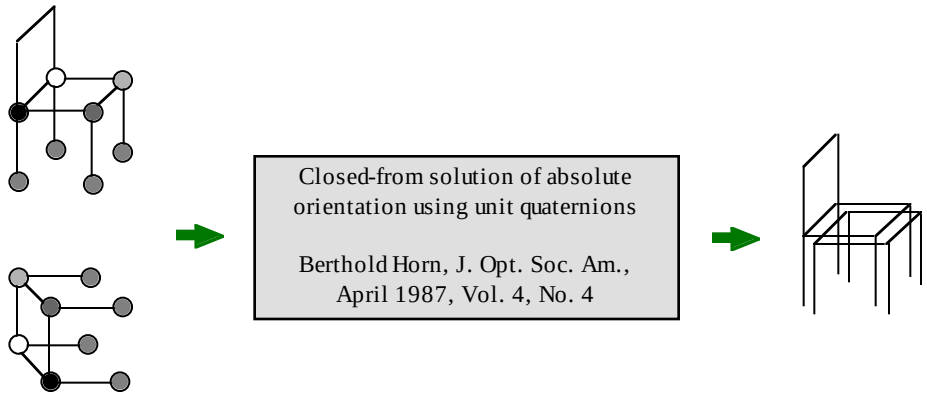

Figure 1. Application of the absolute orientation algorithm to minimize the RMSD between pairs of corresponding points

been able to detect structural similarity between many distantly related proteins. We use both the RMSD value as well as the number of "well aligned" atoms (described below) as criteria for selecting the best superposition. Though we only use the C-α atoms (i.e. the backbones) to represent a structure in our alignment program, the algorithm can be easily extended to include a final step that hierarchically aligns the side-chain atoms as well.

Methods

Overview of the Algorithm

The following three steps briefly describe our hierarchical structural comparison algorithm:

1. ***Local Secondary Structure Superposition***: Compare pairs of vectors from the target and query protein using orientation independent scoring functions (described below). Select the pair that results in the best local secondary structure alignment and transform the query protein to minimize the RMSD between this pair of vectors. Now compare (using dynamic programming) all vectors from the target and query proteins using orientation independent and orientation dependent scores. Transform the query protein to minimize the RMSD between the atoms of the aligned secondary structure elements.
2. ***Atomic Superposition***: For every atom in the query protein, find the nearest atom (within a threshold distance) on the target protein. Transform the query protein to minimize the RMSD between these pairs of atoms. Iterate until the RMSD converges.
3. ***Core Superposition***: Find the best core of correctly aligned and sequentially ordered atoms and minimize the RMSD between them. Iterate until the RMSD converges.

In each of the above steps, the transformation matrix that minimizes the RMSD between pairs of points is computed using the absolute orientation algorithm described in the following section.

Absolute Orientation of Corresponding Points

A fundamental question that needs to be considered in the structural alignment of two objects arbitrarily positioned in the same coordinate system is the following: given a list of pairs of points, where the first point of each pair is taken from object A and the second from object B, is it possible to compute a transformation matrix that, when applied to object B, minimizes the root mean squared deviation between these pairs of points? For example, to align the chair and table in Figure 1 we would select the 8 pairs of corresponding points that are shown in the figure. A closed form solution to this problem exists (Horn, 1987; Horn et al, 1988) which computes, in time proportional to the number of pairs of points, the 4x4 transformation matrix that minimizes the RMSD between these points. Other solutions to this problem have also been proposed (Kabsch, 1978; Arun et al, 1987). In this paper we use the solution presented by Horn, which we shall refer to as the absolute orientation algorithm.

The application of the absolute orientation algorithm is possible only when correspondences between matching points in the two objects are known. In the case of the structural superposition of proteins, these correspondences are not always known. It is possible to try to estimate the correspondences between residues of the two proteins using sequence alignment techniques, but this would lead to highly erroneous results when distantly related proteins are considered that have low sequence identity (Bashford et al, 1987; Lesk and Chothia, 1980). Since it is these very distantly related proteins with low levels of structural similarity that we would like to detect, we developed a structural comparison algorithm that does not use sequence information at any stage.

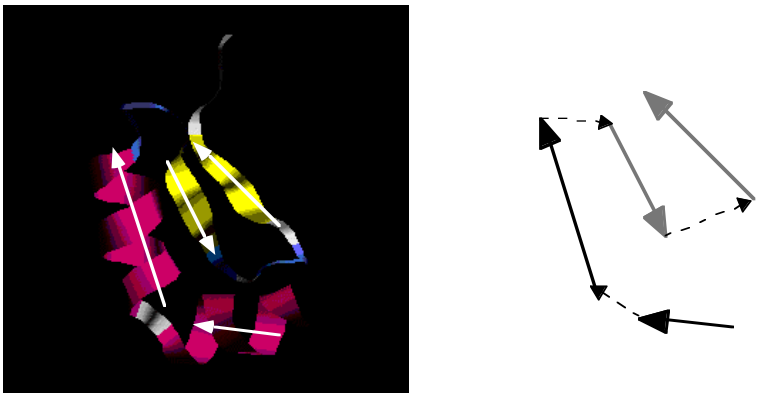

Figure 2. Representing secondary structure elements as vectors

Step 1: Local Secondary Structure Superposition

Our use of secondary structures to find an initial orientation of the two proteins is based on the assumption that it is the well defined secondary structure elements within a protein that provide most of the stability and functionality to the protein and it is therefore these regions that are more conserved through the evolution of the molecule. The secondary structures that we base our alignment on are α-helices and β-strands. In our current implementation all types of helices (α, π, 3-10, and left handed helices) are grouped together in one class. This can easily be altered to create special classes for each type of helix. The classification of residues as either helix or strand was done using the DSSP program (Kabsch and Sander, 1983).

To obtain an initial structural superposition, we first represent each of the helices and strands in the two proteins as individual vectors. For a helix beginning at residue *i* and ending at residue *j*, the following equations are used to compute the beginning and end points of its representative vector:

$$X_{origin} = (0.74*X_i + X_{i+1} + X_{i+2} + 0.74*X_{i+3})/3.48$$
$$X_{end} = (0.74*X_j + X_{j-1} + X_{j-2} + 0.74*X_{j-3})/3.48$$

The multiplying factor of 0.74 in the above equations is due to the fact that the rotation angle between the *i+3* and *i* C-α atoms is 60º while the rotation between the remaining pairs of adjacent atoms (*i* to *1+1*, *i+1* to *i+2*, *i+2* to *i+3*) is 100º. The above weighted sums therefore compute centers of mass that are located at the center of the circles inscribed by the first and last four C-α atoms of the helix (rather than biased to one side). Helices of length less than 4 residues are not considered and helices of length 4 are extended by a single residue on either end to obtain a non-zero representative vector.

Similarly, the following equations are used to compute the beginning and end points of the representative vector for a strand starting at residue *i* and ending at residue *j*:

$$X_{origin} = (X_i + X_{i+1})/2$$
$$X_{end} = (X_j + X_{j-1})/2$$

Having reduced the two proteins to a series of either *H* or *S* vectors (Figure 2), we now use a dynamic programming algorithm to compare these two sets of vectors to find the longest sequence of well matched pairs. The scoring functions used in the algorithm compare single vectors or pairs of vectors from each of the two proteins and return a score that represents the degree of similarity between these vectors. Since we needed to make our computation of the initial alignment independent of the relative orientation and translation of one protein with respect to the other, we defined two sets of scoring functions: orientation independent and orientation dependent. Orientation independent scores are based on the comparison of internal angles and distances between pairs of vectors selected from the two proteins. Orientation dependent scores, on the other hand, compare individual vector orientations and origins from the two proteins. For example, comparing the angle between vector *i* and vector *k* from protein A to the angle between vector *p* and vector *r* from protein B would result in an orientation independent score (Figure 3). On the other hand, comparing the orientation or origin of vector *i* from protein A to the that of vector *p* from protein B would result in an orientation dependent score.

The seven score functions we defined are listed below. The vectors being compared in these functions are those shown in Figure 3. Note that only the vectors shown in solid lines represent secondary structure elements (i.e. *i*, *k*, *p*, and *r*). The dashed lines are the intermediate vectors that join the start and end points of two secondary structure vectors. The sum of the functions listed below gives us the final similarity score between vectors *k* and *r*, with the previous two aligned vectors in the alignment path being *i* and *p*.

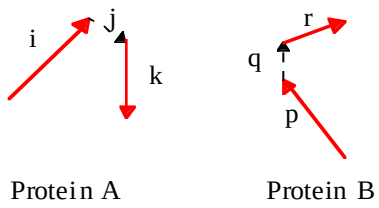

Figure 3. Alignment of secondary structure vectors

Orientation Independent Scores:

$S_1 = S\{|angle(i,k) - angle(p,r)|\}$
$S_2 = S\{|angle(i,j) - angle(p,q)|\}$
$S_3 = S\{|angle(j,k) - angle(q, r)|\}$
$S_4 = S\{|distance(i,k) - distance(p,r)|\}$
$S_5 = S\{|length(k) - length(r)|\}$

Orientation Dependent Scores:

$S_6 = S\{angle(k,r)\}$
$S_7 = S\{distance(k,r)\}$

The function S is defined as follows (Gerstein and Levitt, 1996):

$$S(d) = \frac{2M}{1 + \left[\frac{d}{d_0}\right]^2} - M$$

where, M = maximum score
d = attribute value
d_0 = value at which score should be 0

The distance between two vectors is computed by averaging the distances between the corresponding start, middle, and end points of the vectors. The angle is computed by taking the inverse cosine of the dot product of the two representative unit vectors. The value of the M parameter in the above equations acts as a relative weighting factor for the attribute being measured. Various values of the M and d_0 parameters for the seven score functions were tested based on the relative significance of the attribute being compared and the expected value of the attribute. For example, lengths are de-emphasized by giving them a maximum score of 5 (function S_7) while angle scores are given a maximum score of 10 (function S_6). The final values of the parameters that we used in our program are listed in Table 1. These values were selected by iteratively modifying them and then testing the program with a chosen set protein pairs. We found during this tuning stage that the final output of the algorithm was relatively insensitive to moderate variations in these parameters, though they did affect the ability of the program to detect the small degree of structural similarity retained by distantly related proteins (e.g. myoglobin and colicin).

Table 1. Parameter values for the seven score functions

	S_1	S_2	S_3	S_4	S_5	S_6	S_7
M	10	4	4	2	5	10	5
d_0	45°	45°	45°	5 Å	7 Å	45°	4 Å

We implemented a dynamic programming algorithm, using the scoring functions defined above, to align the secondary structure vectors of the two proteins. Based on the type of score functions used, we were able to obtain either an orientation independent or an orientation dependent alignment. We used a variation of the Smith-Waterman algorithm (Smith and Waterman, 1981) to find the best local alignment of the secondary structure vectors. The gap penalty was set to 0 since we did not want to penalize the deletion of a secondary structure element which could either be due to an incorrect classification by the DSSP program or due to a single mutation that bent a helix or converted a strand to a turn. Our modification to the Smith-Waterman algorithm did not allow the score to decrease at any point during the computation of the score matrix. Therefore, if the score computed for position i,j from position p,q is less than the score of position p,q then the score of position i,j is set to 0. This modification was implemented because we were willing to trade-off the length of the alignment towards greater confidence in the final (shorter) reported alignment. We therefore terminate the alignment when any one pair of vectors returns a negative score (even though the final score for that position might be positive when it is added to the previous best score in the alignment path).

The following algorithm uses the above secondary structure alignment technique to compute an initial superposition that is independent of the relative orientation of the two proteins.

1. Run the dynamic programming algorithm using only orientation independent scores
2. Use the start and end points of all the vectors in the highest scoring alignment to obtain a transformation matrix that best overlaps these points. Apply this matrix to the entire query protein.
3. Refine this initial alignment by repeating steps 1 and 2 using both orientation independent and orientation dependent scores.

The above algorithm may often select an incorrect initial superposition because of the fact that the highest scoring alignment using only orientation independent scores does not necessarily result in the highest scoring alignment when both orientation independent and orientation dependent scores are used. Therefore several of the sub-optimal alignments obtained from the orientation independent step (step 1) also need to be tested in step 3 to find the best initial alignment. To solve this problem we implemented the following algorithm instead:

1. For every pair of vectors in the query protein, find all pairs of vectors in the target protein that align well to this pair of query vectors. The two pairs of vectors are selected by comparing the total score obtained by summing the orientation independent functions, S_1-S_5, to a threshold value. Also, at least one pair of vectors from either the query or the target protein must be adjacent (i.e. not have any gaps).
2. For each set of 4 vectors selected above (2 from the query protein and 2 from the target) find the transformation matrix that best overlaps these vectors. Apply this matrix to the entire query protein.
3. For each initial alignment found in step 2, run the dynamic programming algorithm using both orientation independent and orientation dependent scores. The traceback step to find the alignment is not executed here since only the highest score is needed.
4. Re-apply the transformation from step 2 that produces the highest score in step 3 to the entire query protein.
5. Re-align the two proteins using dynamic programming with both orientation independent and orientation dependent scores.
6. For each pair of aligned secondary structure elements from step 5, find the transformation matrix that minimizes the RMSD between the atoms in the query secondary structure and their nearest neighbors in the aligned target secondary structure.
7. Apply the transformation to all atoms in the query protein and iterate steps 5,6, and 7 until the RMSD converges.

Step 2: Atomic Superposition

Assuming the initial superposition obtained in the previous step is close to the globally optimal alignment, we can now use the following simple three step algorithm to minimize the RMSD between atoms from the two proteins:

1. For every atom in the query protein, find the nearest atom in the target protein.
2. Apply the transformation that minimizes the RMSD between these pairs of atoms to the query protein.
3. Iterate steps 1 and 2 until the RMSD converges.

By iteratively minimizing the RMSD between the two proteins this algorithm essentially performs a greedy descent to the closest local minimum in alignment space. The alignment space here is defined as a six dimensional coordinate system, with three dimensions for the rotation around each axis and three for the translation. The final goal of structural superposition is therefore to find the point in this alignment space that results in the "best" overlap of the two proteins.

During this stage of the algorithm, we only consider those pairs of atoms that are aligned within T Å. The value we selected for T was 3 Å. The algorithm uses this threshold value to distinguish between those atoms that it should or should not include in the computation of the transformation matrix. Since there may be deleted regions or sub-domains in one protein that do not align to any region in the other protein, these deleted regions have to be ignored during this RMSD minimization stage. For example, there are several proteins that contain multiple sub-domains of which only one domain correctly aligns to the target while the other domains constitute deleted or mutated regions. If a threshold value is not used, the algorithm will find an alignment that minimizes the RMSD between all atoms of the two proteins, instead of only minimizing the RMSD between regions that are already close to each other.

Step 3: Core Superposition

While the RMSD value between the two proteins can be considered a criterion for judging the quality of an alignment, it does not result in a fair assessment since the number of aligned atoms considered in the RMSD computation may vary significantly from one alignment to the next. We therefore chose to use both the RMSD value and the number of "well" aligned atoms to measure the quality of an alignment. We implemented the following two tests to determine whether or not an atom is well aligned:

1. For every atom in protein A, find the nearest atom in protein B, and for every atom in protein B, find the nearest atom in protein A (considering only atoms that are aligned within T Å). Select all pairs of atoms that find each other as nearest neighbors. For example, atom i in protein A and atom p in protein B are selected only if p is the nearest neighbor to i and i is the nearest neighbor to p.
2. Delete all pairs of atoms in the list obtained from step 1 that violate co-linearity; i.e., select the maximum number of aligned pairs that are in order in both protein A and B and ignore the rest. This is done by first sorting all pairs of atoms in increasing order based on their atom number in protein A. The resulting sequence of atom numbers from protein B is then parsed to find the maximum number of in order atoms.

The well aligned atoms selected by this technique can be considered the core alignment of the two proteins. It is therefore conceivable that the algorithm should now try to improve the superposition of these core atoms even at the cost of degrading the superposition of all non-core atoms. This final refinement step is implemented by iteratively

determining a new core of atoms and then minimizing the RMSD between these atoms until the RMSD of the core converges.

Analysis of the Algorithm

The complexity of our algorithm can be broken up into two components. The local secondary structure superposition step (step 1) is of *O((max(n,m))*n*m)*, where *n* and *m* are the number of secondary structure elements in the two proteins. The atomic and core superposition steps (steps 2 and 3) are *O(n*m)*, where *n* and *m* are the number of C-α atoms in the two proteins.

The program we implemented to run the above algorithm is both robust and fast. Since we are searching for local alignments in the dynamic programming stage, our method is capable of detecting global similarity between proteins as well as local similarity of sub-domains. We have tested our program by searching a representative set of the PDB using various query proteins and motifs and detected structural similarity between many distantly related proteins. The details and results of these tests are presented in the following section. Each database search compared a single protein against 796 representative proteins from the PDB. The total time required for this search, using sperm whale myoglobin (1mbc, 153 residues) as the query, was 18.28 minutes on a 180 MHz MIPS R5000 microprocessor. Approximately 62% (11.33 minutes) of the execution time for this search was spent on the secondary structure and atomic superposition stages, and the remaining 38% (6.95 minutes) was spent on the core superposition stage.

Results

We compared our structural superposition algorithm, LOCK, to that of Holm and Sander. Their algorithm, DALI, was used to construct the Families of Structurally Similar Proteins (FSSP) database (Holm and Sander, 1996b). As a test of our method, we obtained all the structural neighbors of several proteins, as reported in FSSP, and performed structural comparisons using LOCK of each of these query proteins with all their neighbors. The results for the following three proteins are shown in Figures 4, 5, and 6:

- Sperm whale myoglobin (1mbc)
  153 residues, all alpha
  Number of structural neighbors 152
  Time for search: 2 min, 27 sec

- Triosphosphate Isomerase (1btm-A)
  251 residues, alpha-beta barrel
  Number of structural neighbors 158
  Time for search: 11 min, 14 sec

- Complex - MHC II/Peptide (1seb-A)
  181 residues, mostly beta
  Number of structural neighbors 49
  Time for search: 2 min, 24 sec

The graphs in Figures 4, 5, and 6 plot the protein number along the X-axis (i.e., the number of the protein in the list of structural neighbors reported by FSSP). Figures 4a, 5a, and 6a show the RMSD values computed by DALI and LOCK while Figures 4b, 5b, and 6b show the number of aligned residues (i.e., the number of residues included in the RMSD calculation) reported by DALI and LOCK. The proteins along the X-axis are sorted based on the significance-score for each of the alignments reported in FSSP and not based on the RMSD value. The alignments closest to the origin are therefore considered most significant in the FSSP database. We can see from these graphs that LOCK is able to detect all of the close structural neighbors for each of the query proteins. Since the number of aligned residues varies greatly among the distantly related or unrelated proteins, we cannot use the RMSD value to compare our results with those produced by DALI. For example, the RMSD values reported by LOCK never exceed 3 Å due to the threshold value we enforced which only considers those atoms that are within this range. LOCK therefore yields alignments with lower RMSD values for distantly related proteins because it classifies fewer atoms as being correctly aligned. The curves for the number of aligned atoms found by LOCK show a very distinct cut-off beyond which the two proteins can be considered un-related. For the case of myoglobin, the proteins close to this cut-off value (approximately 100-120 on the X-axis) included those proteins with small degrees of structural similarity to myoglobin, such as the phycocyanins and colicin.

We also tested our program by performing a series of PDB searches using both complete proteins and small structural motifs as queries. The representative set of the PDB that we searched contained a maximal set of proteins that had a resolution of less than 3.5 Å and sequence similarity of less than 40%. The number of proteins found, using the OBSTRUCT algorithm (Heringa et al, 1992), were 796. The two query structures whose results are reported here are the following:

- Sperm whale myoglobin (1mbc)
  153 residues, all alpha
  Time for search: 18 min, 17 sec

- Helix-Turn-Helix motif
  Residues 32-53 from 1lmb-3
  Time for search: 3 min, 14 sec

The results of these searches were sorted according to the number of well aligned atoms. The top 25 hits for the above query structures are shown in Tables 2 and 3.

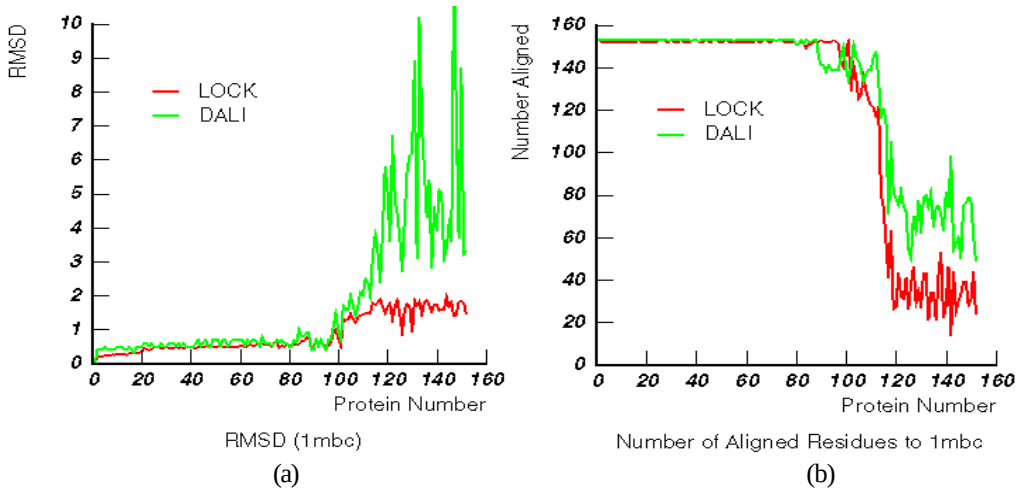

Figure 4. Comparison of LOCK and DALI - Sperm whale myoglobin

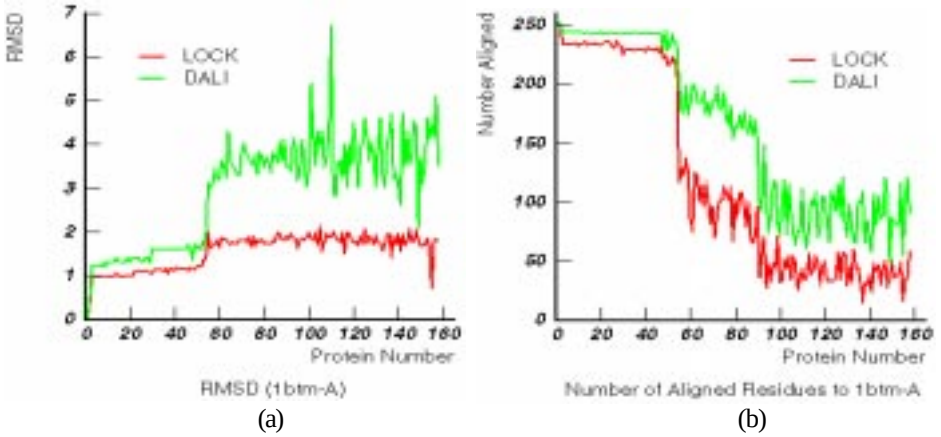

Figure 5. Comparison of LOCK and DALI - Triosphosphate Isomerase

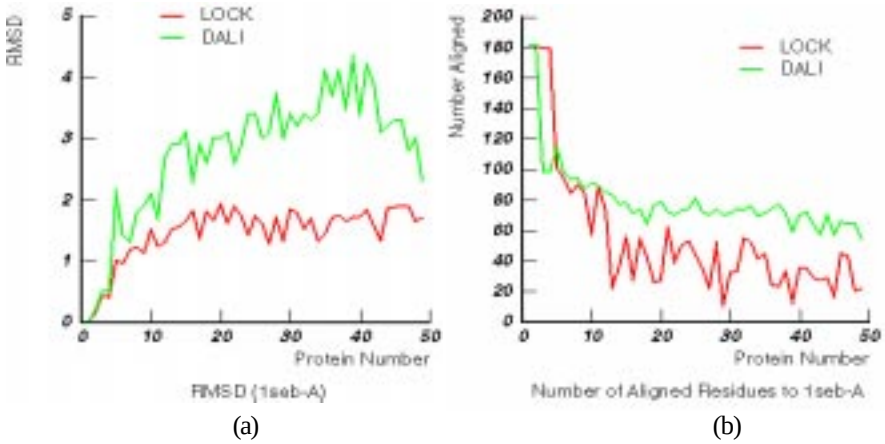

Figure 6. Comparison of LOCK and DALI - Complex-MHC II/Peptide

Table 2. Top 25 hits obtained by searching the PDB using sperm whale myoglobin as the query

Protein	RMSD	Atoms Aligned	PDB Header
1myh-A	0.562531	153	MYOGLOBIN (AQUOMET, PH 7.1) MUTANT
2dhb-A	1.335804	132	HEMOGLOBIN (HORSE,DEOXY)
1eca	1.497830	130	HEMOGLOBIN (ERYTHROCRUORIN, AQUO MET)
1flp	1.349387	129	HEMOGLOBIN I (MONOMERIC) (FERRIC)
2lhb	1.084818	128	HEMOGLOBIN V (CYANO,MET)
1hds-B	1.235697	128	HEMOGLOBIN (SICKLE CELL)
1ith-A	1.302358	126	HEMOGLOBIN (CYANOMET)
1mba	1.501337	126	MYOGLOBIN (MET) ($P*H 7.0)
1hbg	1.412812	125	HEMOGLOBIN (CARBON MONOXY)
1ash	1.431599	122	HEMOGLOBIN (DOMAIN ONE)
1hlb	1.475264	121	HEMOGLOBIN (SEA CUCUMBER)
1hbi-A	1.408008	118	HEMOGLOBIN I (OXYGENATED, HOMODIMER)
1gdi	1.476536	118	LEGHEMOGLOBIN (CARBON MONOXY)
1cpc-A	1.787059	80	C-PHYCOCYANIN
1cpc-L	1.682854	74	C-PHYCOCYANIN
1tox-A	1.640087	63	DIPHTHERIA TOXIN DIMER COMPLEXED WITH NAD
1pbg-A	1.701624	60	MOL_ID: 1;
1col-A	1.798676	58	COLICIN *A (C-TERMINAL DOMAIN)
2sbl-B	1.519786	53	LIPOXYGENASE-1 (SOYBEAN)
1oxa	1.851156	52	CYTOCHROME P450 (DONOR:O2 OXIDOREDUCTASE)
1krb-C	1.680887	49	MOL_ID: 1;
1irk	1.409630	48	INSULIN RECEPTOR (TYROSINE KINASE DOMAIN)
2hpd-A	1.619509	48	CYTOCHROME P450 (BM-3)
1le2	1.623706	48	APOLIPOPROTEIN-*E2
2cp4	1.900296	47	CYTOCHROME P450CAM (CAMPHOR MONOOXYGENASE)

Table 3. Top 25 hits obtained by searching the PDB using the helix-turn-helix motif as the query

Protein	RMSD	Atoms Aligned	PDB Header
1lmb-4	0.207748	22	DNA-BINDING REGULATORY PROTEIN
1pra	0.451552	22	GENE REGULATING PROTEIN
1adr	0.472226	22	TRANSCRIPTION REGULATION
1yrn-A	0.492206	22	COMPLEX (TWO DNA-BINDING PROTEINS/DNA)
1dik	0.517499	22	PHOSPHOTRANSFERASE
1pnr-A	0.613499	22	COMPLEX (DNA-BINDING REGULATION/DNA)
4fis-A	0.677586	22	DNA-BINDING PROTEIN
1oct-C	0.697683	22	DNA-BINDING PROTEIN
1ftt	0.742871	22	DNA BINDING PROTEIN
1cop-E	0.768256	22	GENE REGULATING PROTEIN
1dtr	0.780090	22	DNA BINDING PROTEIN
1gdt-A	0.795171	22	COMPLEX (SITE-SPECIFIC RECOMBINASE/DNA)
3gap-B	0.817812	22	GENE REGULATORY PROTEIN
1lfb	0.820489	22	TRANSCRIPTION REGULATION
1ads	0.920977	22	OXIDOREDUCTASE
1pdn-C	0.949104	22	COMPLEX (GENE REGULATING PROTEIN/DNA)
1hom	0.958631	22	DNA-BINDING PROTEIN
1mse-C	0.965817	22	COMPLEX (BINDING PROTEIN/DNA)
1cma-A	1.023169	22	DNA-BINDING REGULATORY PROTEIN
1trr-A	1.048990	22	DNA-BINDING REGULATORY PROTEIN
1tpl-B	1.467803	22	LYASE(CARBON-CARBON)
1hcr-A	0.930009	21	DNA-BINDING
1neq	0.949148	21	DNA-BINDING PROTEIN
1ade-A	0.979226	21	LIGASE (SYNTHETASE)
153l	1.093427	21	HYDROLASE(O-GLYCOSYL)

Discussion

The algorithm we have developed for protein structure superposition is based on alignments at both the secondary structure level and the atomic level. This use of secondary structure information gives our technique the increased flexibility of detecting global as well as local similarities. By first comparing only pairs of vectors from both proteins, we are able to find a short sequence of well aligned secondary structure elements that serves as an anchor for the remaining RMSD minimizing and refinement stages of the algorithm. Our representation of secondary structures as single vectors enables the use of multiple vector comparison techniques and also significantly reduces the computation required to find a good initial superposition of the two proteins. The initial superposition is then improved by iteratively applying the absolute orientation algorithm to minimize the RMSD between all pairs of nearest atoms from both proteins. This stage of the algorithm searches the alignment space to find the local minimum closest to the initial superposition obtained from the previous step. It is this hierarchical technique of first searching among secondary structure alignments and then among atomic level alignments that results in the increased flexibility and speed of our algorithm.

We implemented our algorithm on a Silicon Graphics Indy workstation (MIPS R5000, 180 MHz) and provided a convenient graphical user interface (Tcl-Tk based) to run the alignment and view the structural superposition of the two proteins. The program takes approximately 18 minutes to compare myoglobin (153 residues) to a representative set of 796 proteins from the PDB. Our alignment program also performs iterative subdivisions and re-alignments of the non-aligned regions from the two proteins. In addition, when two sub-domains are found in the query protein that align separately to two different regions of target, the program computes the optimal bending point in the query protein that will maximize the number of aligned atoms.

The results of a comparison of our alignment technique to that of Holm and Sander show that we are able to detect structural similarities at the same level as those detected by their DALI algorithm. In addition, by searching a representative set of the PDB using myoglobin, we were able to detect a few proteins with low structural similarity that were not included in the list of proteins that were reported as structurally similar to myoglobin in the FSSP database. We also used our program to align the pairs of proteins that were used as test cases in the paper by Falicov and Cohen on the minimum area metric for structural comparisons (Falicov and Cohen, 1996). Though we did not perform a detailed analysis of the alignments reported by both techniques, our program found alignments that were in all cases close to those of Falicov and Cohen. We tested the ability of our program to detect small structural motifs within proteins by searching a representative set of the PDB using the helix-turn-helix and EF-hand motifs as query structures. The results for the helix-turn-helix motif (Table 3) show that the program correctly ranked the proteins that contained this motif (mostly DNA binding proteins) at the top of the list.

LOCK will be made available to academic and non-profit institutions by request to the authors. Commercial firms should contact the Office of Technology Licensing at Stanford University.

Acknowledgments

This work was supported by the NLM R01 LM05716-01 grant. The authors thank Russ Altman, Thomas Wu, Barry Robson, Brian Curless, Bernd Froehlich, and Tamara Shannon for their valuable discussions and editing assistance.

References

Arun, K.S., Huang, T.S., and Blostein, S.D. 1987. Least-squares fitting of two 3-D point sets. IEEE *Transactions on Pattern Analysis and Machine Intelligence* 9(5):698-700.

Barakat, D.W. and Dean, P.M. 1991. Molecular structure matching by simulated annealing, III. The incorporation of null correspondences into the matching problem. J. Comp.-Aided Mol. Design 5:107-117.

Bashford, D., Chothia, C, and Lesk, A.M. 1987. Determinants of a protein fold: Unique features of the globin amino acid sequences. *J. Mol. Biol*. 196:199-216.

Bernstein, F.C., Koetzle, T.F., Williams G.J.B., Meyer E.F. Jr., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T., and Tasumi, M. 1977. The Protein Data Bank: A computer based archival file for macromolecular structure. *J. Mol. Biol.* 112:535-542.

Chothia, C. 1992. One thousand folds for the molecular biologist. *Nature* 257:543-544.

Falicov, A and Cohen, F.E. 1996. A surface of minimum area metric for the structural comparison of proteins. *J. Mol. Biol.* 258: 871-892.

Gerstein, M and Levitt, M. 1996. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. In Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology, 59-67. Menlo Park, Calif.: AAAI Press.

Godzik, A. and Skolnick, J. 1994. Flexible algorithms for direct multiple alignment of protein structures and sequences. *CABIOS* 10:587-596.

Heringa, J., Sommerfeldt, H., Higgins, D., and Argos, P. 1992. OBSTRUCT: A program to obtain largest cliques from a protein sequence set according to structural resolution and sequence similarity. *CABIOS* 8(6):599-600.

Holm, L. and Sander, C. 1996a. Mapping the protein universe. *Science* 273:595-602.

Holm, L. and Sander, C. 1996b. The FSSP database: fold classification based on structure-structure alignment of proteins. *Nucleic Acids Research* 24(1):206-209.

Holm, L and Sander, C. 1995. 3-D Lookup: Fast Protein Structure Database Searches at 90% Reliability. In Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology, 179-187. Menlo Park, Calif.: AAAI Press.

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

Horn, B.K.P. 1987. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am.*, 4(4):629-642.

Horn, B.K.P., Hilden, H.M., and Negahdaripour, S. 1988. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am.*, 5(7):1127-1135.

Kabsch, W. 1978. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. A* 34:827-828.

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

Lesk, A.M. and Chothia, C. 1980. How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. *J. Mol. Biol.* 136:225.

Mitchell, E.M., Artymiuk, P.J., Rice, D.W., and Willett, P. 1989. Use of techniques derived from graph theory to compare secondary structure motifs in proteins. *J. Mol. Biol.* 212:151-166.

Orengo, C.A. and Taylor, W.R. 1996. SSAP: Sequential Structure Alignment Program for protein structure comparison. *Meth. in Enzym.* 266:617-635.

Orengo C.A., Flores, T.P., Taylor, W.R., and Thornton J.M. 1993. Identification and classification of protein fold families. *Protein Eng.*, 6:485-500.

Russel, R.B. and Barton, G.B. 1993. Multiple protein sequence alignment from tertiary structure comparisons: Assignment of global and residue confidence levels. *Proteins: Struct. Funct. Genet.* 14:309-323.

Sali, A. and Blundel, T. 1990. 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.* 212:403-428.

Smith, T.F. and Waterman, M.S. 1981. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195-197.

Subbiah, S., Laurents, D.V., and Levitt, M. 1993. Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core. *Current Biology* 3(3):141-148.

Taylor, W. and Orengo, C. 1989. Protein structure alignment. *J. Mol. Biol.* 208:1-22.

Vriend, G. and Sander, C. 1991. Detection of common 3-D substructures in proteins. *Proteins* 11:52-58.

Zuker, M. and Somorjai, R.L. 1989. The alignment of protein structures in three dimensions. *Bull. Math. Biol.* 51(1):55-78.