Protein Structure Alignment: A Comparison of Methods

Amit P. Singh*
Section on Medical Informatics
Stanford University
Stanford, CA 94305, USA

Douglas L. Brutlag
Section on Medical Informatics and Department of Biochemistry
Stanford University
Stanford, CA 94305, USA

Running Head: Protein Structure Alignment

Keywords: protein structure alignment, fold identification, sensitivity, specificity

*To whom correspondence should be addressed. Address: Beckman Center B400, Department of Biochemistry, Stanford University, Stanford, CA 94305-5307.
Tel: 650-723-5976. Fax: 650-725-6044. E-mail: apsingh@cmgm.stanford.edu

Abstract

Motivation: Algorithms for the alignment of protein structures have grown increasingly important with the recent and rapid growth of the protein structure database. Several techniques are currently available that attempt to find the optimal alignment of shared structural motifs between two proteins. Our goal in this paper is to quantitatively compare the sensitivity and specificity of some of these structural alignment techniques. We use the *scop* (Structural Classification of Proteins) database [Murzin et al., 1995] as the 'gold standard' for this evaluation. For a given query structure, the *scop* classification is used to determine the structurally related (true positive) and unrelated (true negative) structures from a list of 685 representative target structures.

Results: The sensitivity and specificity of each program is computed by sorting the list of alignments they generate in descending order and counting the number of true positive and false positive structures above every position in the list. The results of this study are plotted on ROC-like curves (Receiver Operator Characteristic) for each query structure. We also examine the relative computational efficiency of the different techniques. The results show that of the six algorithms included in the evaluation, DALI [Holm and Sander, 1993a] and LOCK [Singh and Brutlag, 1997] yield the best performance in terms of sensitivity and specificity. In terms of speed, LOCK ranks higher than all programs that compute alignments at the level of amino acids.

Contact: apsingh@cmgm.stanford.edu

Introduction

Protein structure alignment techniques have grown increasingly important as a means to quantitatively compare and classify all known protein structures. The number of structures in the Protein Data Bank (Bernstein et al., 1977) is currently (as of March 1999) more than 9,400, with almost 150 new structures being added every month. The number of known fold families into which these structures can be classified, are, on the other hand, relatively few (Chothia, 1992; Orengo et al., 1993; Murzin et al., 1995; Holm and Sander, 1996a; Brenner et al., 1997). One of the primary goals of structural alignment programs is to quantitatively measure the level of structural similarity between all pairs of known protein structures. This data can provide several meaningful insights into the nature of protein structures and their functional mechanisms. For instance, the comparison of all structures against each other can show relationships, both functional and structural, between proteins that were previously not known to be related (Holm and Sander, 1993b, Artymiuk et al., 1997; Bryant et al., 1997). In addition, structure based distance measures are critical to constructing accurate phylogenies of proteins and classifying structures into families that share similar folds or motifs. Identifying these shared structural motifs using structural alignment techniques can provide significant insight into the functional mechanisms of the protein family.

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 alignment of secondary structure elements as well as alignment of intra and inter-molecular atomic distances (Mitchel et al., 1989; Zuker and Somorjai, 1989; Taylor and Orengo, 1989; Sali and Blundel, 1990; Vriend and Sander, 1991; Barakat and Dean, 1991; Russel and Barton, 1992; Subbiah et al., 1993; Holm and Sander, 1993a; Godzik and Skolnick, 1994; Holm and Sander, 1995; Orengo and Taylor, 1996; Falicov and Cohen, 1996; Gerstein and Levitt, 1996; Singh and Brutlag, 1997).

Our objective in this paper is to quantitatively compare some of the above structure alignment algorithms in an effort to better understand their various advantages and limitations. An obvious difficulty in performing such an evaluation is selecting a standard against which to make each of the comparisons. Because of the inherent difficulty of identifying and aligning the key structural motifs that characterize a family of proteins, it is often unclear which proteins from the database should be classified as true members of that family. The manually constructed *scop* (Structural Classification of Proteins) database (Murzin et al., 1995) has often been cited as a

possible 'gold-standard' for structural alignment methods. The *scop* database has been constructed by visual inspection of all structures in the Protein Data Bank (PDB). The four levels of the *scop* hierarchy are: 'Class', 'Fold', 'Superfamily', and 'Family'. Structures within the same 'Family' generally have significant sequence similarity and hence show a clear evolutionary relationship. Structures in the same 'Superfamily' have a high degree of structural similarity and are considered, by the developers of the database, to probably have a common evolutionary origin. Structures within the same 'Fold' are those that have major structural similarity but not necessarily a common evolutionary origin. Finally, structures within the same 'Class' are those that have similar overall secondary structure content. Since the *scop* database has been developed manually, it avoids the problems of automated alignment and classification techniques but at the obvious cost of introducing the biases and limitations of the experts' knowledge. Nevertheless, the database is being widely used and has been recognized as a current standard in structural classification. For our purposes we use the *scop* classification to find all proteins that should be considered structurally related to a given query structure.

Our approach to comparing structural alignment programs is based on evaluating the sensitivity and specificity of each method using *scop* as the gold standard. We examine the ability of each alignment method to correctly identify all structures that belong to same fold as a given query structure. In addition, where possible, we also examine the computational efficiency of each technique. The structural alignment algorithms that we compare in this paper are:

1. DALI (Holm and Sander, 1993a)
2. STRUCTAL (Gerstein and Levitt, 1996)
3. VAST (Gibrat et al., 1996)
4. MINAREA (Falicov and Cohen, 1996)
5. LOCK (Singh and Brutlag, 1997)
6. 3dSEARCH (Singh and Brutlag, 1998, unpublished)

These techniques are described briefly in the Methods section.

Methods

The primary objective of our comparison of structural alignment programs is to examine the sensitivity and specificity of each method. To perform this comparison, we obtained a representative subset of the PDB such that all pairs of structures in the subset have less than 25% sequence similarity (using PDB-Select (Hobohm and Sander, 1992)). We then used each program to align three query structures to all of the 685 structures in this representative target set and sorted the alignments based on the similarity scores reported by each method. For each query structure, we classified the 685 target structures as being either structurally similar (true positive) or unrelated (true negative) using version 1.37 of the *scop* database. Those structures that belonged to the same *scop* Fold as the query structure were classified as structurally similar true positives and the remaining structures were classified as true negatives. Since version 1.37 of *scop* (which is the currently available version, as of March 1999) was released in October 1997, we used a PDB-select list that was generated before October 1997 to ensure that all chains in the list were represented in the *scop* database.

To compare the performance of the various techniques, we plotted the number of true positives vs. the number of false positives for each method. The data points for the graphs were computed by incrementally descending the sorted lists produced by each alignment program and counting the number of true positives and false positives above every position in the list. The curves obtained are, therefore, similar in nature to the traditional ROC (Receiver Operator Characteristic) curves that plot sensitivity on the vertical axis and 1-specificity on the horizontal axis. In addition, for each of the three query structures, we report the true positives and false negatives found by each program at a specificity cutoff of 90%. We used the following three query structures for our comparison:

1mbd Sperm whale myoglobin; 153 residues

scop Class: All-α

scop Fold: Globin-like

1tph-2 Triose phosphate isomerase, chain 2; 245 residues

scop Class: α and β (α/β)

scop Fold: β/α (TIM) barrel

8fab-A Immunoglobulin, chain A, constant domain; residues 106-208

scop Class: All-β

scop Fold: Immunoglobulin-like β sandwich

The above query structures were selected because they represent fundamentally different structural topologies that cover the three most prominent classes of protein structures.

The following sections describe the six structural alignment algorithms and how they were used in this study.

DALI

DALI is based on the alignment of 2-dimensional distance matrices, which represent all intra-molecular Cα-Cα distances of a protein structure (Holm and Sander, 1993a; Holm and Sander, 1996a). For a given pair of structures, DALI attempts to compute the optimal arrangement of similar contact patterns from their respective distance matrices. Each distance matrix is first split into hexapeptide fragments and all pairs of similar fragments from the two structures are stored in a pair list. The final alignment is computed by assembling pairs of overlapping fragments from the pair list. Since computing the optimal arrangement of fragments would require searching an exponentially large search space (Lathrop, 1994), DALI uses the branch-and-bound algorithm (Lathrop and Smith, 1996) to find an approximate solution to this problem. The branch-and-bound algorithm iteratively decomposes the search space into smaller subsets and computes the upper bound of the evaluation function for each subset. The next iteration of the algorithm repeats this process on the subset with the highest upper bound. Since the order of the hexapeptide fragments is not used to limit the search space, the alignments found by DALI do not require the matched substructures to be linearly equivalent in the two structures (though sequential equivalence can be imposed as a constraint). The scoring function for an alignment of two structures is based on the intra-molecular distances between each pair of aligned residues. DALI uses the following elastic similarity measure to compute the score of each pair of aligned residues:

$$score(i, j) = 0.2 - \frac{| distance^{A}(i, j) - distance^{B}(i, j) |}{distance\ (i, j)}\ w(distance\ (i, j))$$

where *i* and *j* each represent a pair of aligned residues between proteins A and B, $distance^{*}(i,j)$ is the average of $distance^{A}(i,j)$ and $distance^{B}(i,j)$, and *w* is an envelope function that gives lower weights to residues that are further apart, thus reducing their relative contribution to the total score. This score is summed over all pairs of aligned residues from both structures. The elastic similarity score used by DALI allows greater variability between residues that are further apart and also reduces their contribution to the total score.

DALI is available at http://www2.ebi.ac.uk/dali/. The DALI server allows the user to perform either pair-wise alignments of single structures or to search a representative set of the PDB (Holm and Sander, 1996b). Results are returned to the user by email and are sorted according to the Z-score obtained by each alignment. Only alignments with a Z-score greater than 2.0 are returned by the server. DALI has also been used to build the FSSP database (Holm and Sander, 1996c). The DALI alignments reported in this paper were computed by the DALI developers on their own computer with linear equivalence imposed as a constraint.

STRUCTAL

STRUCTAL uses iterative dynamic programming to minimize the RMSD between two protein backbones (Subbiah et al. 1993; Gerstein and Levitt, 1996, Gerstein and Levitt, 1998). It is derived from the ALIGN program developed by Cohen (Satow et al., 1987; Cohen, 1997). Dynamic programming is applied to the backbone atoms by first computing pair-wise inter-atomic distances between all C-α atoms from the two structures. Hence, for each C-α atom in the query structure, its distance to all C-α atoms in the target structure is computed. This matrix of pair-wise distances is converted into a scoring matrix using the following equation:

$$score(i,j) = \frac{M}{1 + \sqrt[3]{\left(\frac{distance(i,j)}{d_0}\right)^2}}$$

where *M* is the maximum score given to a match and d_0 is the distance at which the similarity score should be half the maximum value (d_0 = 2.24 Å (Gerstein and Levitt, 1998)). Dynamic programming is used to compute an optimal alignment from this score matrix (Needleman and Wunsh, 1971). The resulting atomic correspondences are used to transform the target structure such that the RMSD between the aligned atoms is minimized. The distance matrix is then computed again and this process is iterated until convergence. Since this iterative dynamic

programming algorithm is dependent on the initial orientation of the two structures, several different starting configurations are used and the final alignment with the best score is selected. The six different starting alignments reported in Gerstein and Levitt, 1998 are: (1) align the beginnings of the two sequences, (2) align the midpoints of the two sequences, (3) align the ends of the two sequences, (4) align at a random point, (5) align using sequence similarity, and (6) align using alpha angles. This paper also describes several enhancements to STRUCTAL's original alignment procedure. These include using side-chain orientation to avoid single-residue misalignments, using exposure weighting to increase the significance of aligned residues buried inside the structure, using secondary structure-dependent gap penalties, and splitting certain structures into smaller fragments.

STRUCTAL is available at http://bioinfo.mbb.yale.edu/Align/server.cgi. We used this server to compute the alignments and sorted them based on the "sscore-p-value" reported by the program (Levitt and Gerstein, 1998).

VAST

VAST is based on aligning secondary structure elements using an algorithm from the field of graph theory (Madej et al., 1995; Grindley et al. 1993). All pairs of secondary structure elements (one from each structure) that have the same type are represented as nodes of a graph. Two nodes are connected by an edge if the distance and angle between the corresponding pairs of secondary structure elements from the two proteins are within some threshold. The graph therefore represents correspondences between pairs of secondary structure elements that have the same type, relative orientation, and connectivity. This correspondence graph is then searched to find the maximal subgraph such that every node in the subgraph is connected to every other node in the subgraph and is not contained in any larger subgraph with this property. This is referred to as clique detection in graph theory and is basis of finding the initial secondary structure alignment. VAST extends this initial alignment to a residue level alignment using a Gibbs sampling technique.

VAST places considerable emphasis on defining the statistical significance of an alignment. For each pair-wise alignment, the algorithm computes an alignment score as well as a P-value for the best substructure superposition. The P-value assigned to the alignment is calculated as the probability that its score would be seen by chance in drawing secondary structure pairs at random from the database multiplied by the number of possible alternative substructure alignments for the given pair of structures. The program only reports alignments that yield a

P-value less than 0.05. A P-value of 0.05 indicates that VAST expects to find an alignment with the same degree of similarity by chance in 5% of all pair-wise comparisons. VAST uses a threshold of 0.05 to limit the noise in the hit lists, thus allowing repeated iterations of "double neighboring" in Entrez (Schuler et al., 1996).

VAST has been used to compare all known PDB domains to each other. The results of this computation are included in NCBI's Molecular Modeling Database at http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.html. We obtained the list of all structural neighbors for each of the three query structures from this site and removed from these lists any structures that were not part of our subset of 685 target structures. We assumed that structures that appeared in our target set but not in the VAST lists had a P-value of lower than 0.05 and were hence not returned by their server. The resulting lists were sorted according to their alignment score and used to compute the sensitivity and specificity of this program.

MINAREA

The MINAREA program computes a triangulation between the C-α atoms of the two proteins in order to minimize the stretched surface area between their backbones. The algorithm uses dynamic programming to find the triangulation between the C-α atoms that would lead to the minimal area. The resulting correspondences are used to transform the structures such that the stretched surface area is minimized. The dynamic programming step is then repeated with these new coordinates and this process is iterated until convergence. This iterative algorithm is seeded with an initial superposition that is based on finding the best RMSD fit between the smaller protein and a sequential subset of the larger protein with the same number of residues. For example, if the smaller protein has N residues, the initial alignment is found by first computing the RMSD between the smaller structure and the first N residues of the larger structure. The smaller structure is then moved forward by a single residue and the RMSD is computed again. This process is repeated until the end of larger protein is reached. The alignment that yields the lowest RMSD value is selected as the initial alignment for the area-minimization algorithm. MINAREA determines the quality of the alignment by computing the Area Functional (AF) which is the minimal area divided by the average length of the two proteins. The program also reports a measure of significance called the Fit Comparison (FC) which is the ratio of the AF to the average distance between any two residues, one from each protein.

We obtained MINAREA from the Cohen Group and ran it locally using the default parameters. We compared each query structure to all structures in the target database and sorted the list of alignments based on their FC ratios.

LOCK

The LOCK algorithm attempts to find the optimal rigid-body superposition of two structures such that the RMSD between the aligned C-α atoms in minimized. Since the correct selection of aligned residues is critical to the success of this approach, LOCK uses an iterative technique of selecting pairs of corresponding residues and then minimizing the RMSD between them. This iterative approach essentially performs a greedy search to the nearest local minimum in alignment space, and hence the algorithm must be seeded with a good initial superposition of the two structures. This initial superposition is computed by aligning secondary structure vectors using dynamic programming. The three main steps of the LOCK algorithm are:

1. *Local Secondary Structure Superposition*

   The secondary structure elements of both proteins are represented as vectors and dynamic programming is applied to find the optimal local alignment of these vectors. The score matrix for the dynamic programming is computed based on a combination of orientation independent and orientation dependent scoring functions. The start and end points of the aligned vectors is used to compute an initial superposition of the structures.

2. *Atomic Superposition*

   Corresponding residues between the two structures are determined by finding, for each C-α atom on the query structure, the nearest C-α atom on the target structure. Only pairs of atoms that are within a certain cutoff distance (usually 3 Å) are included. The RMSD between these pairs of corresponding atoms is minimized (Horn, 1987) and the process is repeated until the RMSD converges.

3. *Core Superposition*

   This step of the algorithm differs from the previous step only in the criteria used to select pairs of corresponding atoms. In this step, only those pairs of atoms are included in the alignment that are mutually found as nearest neighbors. The list of corresponding atoms is also parsed to find the maximal subset of

correspondences that are sequentially ordered in both structures. The RMSD between these selected pairs is minimized and the process repeated until the RMSD converges.

The core superposition step (step 3) limits the final alignment to include only those residues that can be clearly considered to be part of a shared structural motif. Hence, the number of aligned core residues reported by LOCK usually drops significantly for unrelated structures. Since all aligned residues are guaranteed to be within 3 Å, the RMSD is not considered in the final ranking of the alignments. Instead, the number of aligned core residues is used as the principle criterion to determine the significance of the LOCK alignments.

LOCK is available at http://gene.stanford.edu/lock/. The WWW interface provides three primary functions: pair-wise alignment of single structures, alignment of a query structure to a specified list of target structures, and alignment of a query structure to various representative subsets of the PDB. The results of the alignments are sorted based on the number of aligned core residues and can either be displayed on the web browser or returned to the user via email. Execution times for typical pair-wise alignments are in the order of milliseconds to seconds.

3dSEARCH

The 3dSEARCH algorithm is designed to compute fast but approximate alignments of protein structures based on secondary structure elements alone. The algorithm is based on the concept of geometric hashing, developed in the field of computer vision (Lamdan and Wolfson, 1988), and is similar to the program 3d-Lookup developed by Holm and Sander (1995). The fundamental idea of this technique is to represent all secondary structure vectors from all target proteins in a large, highly redundant hash table (or index table). Once the table is built, each secondary structure vector from a given query structure can be simultaneously compared to the entire library of target structures simply by indexing into this table. The algorithm for building the hash table consists of the following steps (executed on all target structures):

1. For each pair of vectors (i, j) in the structure, compute a coordinate system using the start and end points of vector i and the orientation of vector j.
2. Compute the coordinates and orientation of all remaining vectors in the protein based on the coordinate system defined in step 1.

3. Place an entry for each vector from step 2 into the hash table at the location specified by the coordinates of midpoint of the vector. The hash table can be visualized as a large 3-dimensional grid, with each cube of the grid corresponding to a particular bin in the table. Each vector is placed in the bin in which its mid-point lies. The resolution of the grid (i.e. the cube edge length) corresponds to the desired tolerance in each vector alignment. Our current implementation uses a grid size of 2 Å. Each entry in the hash table contains a pointer to the protein, a pointer to the particular coordinate system from which the coordinates of the current vector were computed, and the orientation of the vector in this coordinate system.

A target structure with N secondary structure vectors generates a total of $N*(N - 1)*(N - 1)$ entries in the hash table, with each vector being represented $(N – 1)*(N - 1)$ times.

To compare a given query structure to all target structures, the above steps are repeated for the query with the exception that step 3 is modified to vote for similar target vectors in each bin (instead of adding more vectors to this bin). For each query vector in each coordinate system, the bins accessed by the vector in step 3 are retrieved and votes are given to those target vectors found in these bins whose orientation is similar to the query vector. Hence, instead of adding a new entry to a bin, the previous entries in the bin (as well as all its neighboring bins) are retrieved and compared to the query vector. Votes are given to a coordinate system of a target protein if a target vector in that coordinate system lies in the same bin as the query vector (or in a bin adjacent to it) and has the same orientation and class (helix or strand) as the query vector. Once this is completed for all possible query coordinate systems and vectors, the votes obtained by each target structure are tallied. The target structures are finally sorted based on the maximum number of votes received in any of its coordinate systems. Each vote represents a match between a particular target and query vector and can therefore be used to compute a transformation matrix that minimizes the RMSD between the aligned vectors. Since only secondary structure vectors are compared, 3dSEARCH does not compute a residue-level alignment. Also, since the algorithm does not impose any ordering constraints, the alignments are not restricted to linearly equivalent secondary structure elements from the two proteins.

3dSEARCH is available at http://gene.stanford.edu/3dsearch/. The WWW interface provides four functions: pair-wise alignment of single structures, alignment of a query structure to a specified list of target structures, alignment of a query structure to various representative subsets of the PDB, and alignment of a query structure to the entire

PDB. The full-PDB search compares a given query structure to about 13,000 PDB domains defined by the *scop* database (release 1.37). Since the algorithm is very fast, the results of the alignments are returned directly on the web browser and not sent to the user through email. Execution times for comparing a typical query structure to the 13,000 PDB domains range from about 30 seconds to 3 minutes, depending on the size of the query structure and the load on the server. Hyper-text links allow each vector based alignment to be refined by performing a more detailed atomic alignment using LOCK.

Results

The ROC curves for the three query structures are shown in Figure 1. In addition, the results are summarized in *Figure 1* Tables 1, 2, and 3, since some details may not be visible in the graphs. The missing entries in these tables ("---") *Tables 1,2,3* are due to the significance thresholds of the corresponding programs. Both DALI and VAST terminate the list of reported alignments at a predetermined significance threshold and do not return alignments that scored below this cutoff. Hence, the missing entries correspond to some true positive structures that were given very low significance scores by DALI or VAST and were not returned by the servers for these programs.

To further differentiate between the various techniques, we also tabulated the true positive hits found by each program at a specificity of 90%. Tables 4, 5, and 6 show the true positives (+) and false negatives (-) for each *Tables 4,5,6* of the three query structures.

The execution times for searching the set of 685 target structures are shown in Table 7. The LOCK, *Table 7* 3dSEARCH, and MINAREA searches were run locally on a Silicon Graphics Octane with a 195 MHz MIPS R10000 processor. Since the DALI searches were performed by the DALI developers on their own computer (also a 195 MHz R10000 processor), the execution times reported for this program are estimates provided by them. The execution times for DALI (in Table 7) do not necessarily correspond to those of the DALI server which uses a hierarchy of preprocessing steps, as well as a database of precomputed alignments, to decrease the processing time for each request. The STRUCTAL alignments were computed on a remote server through the internet and times reported have therefore been adjusted to account for network latencies. We were not able to include execution times for VAST since the alignments for this program were downloaded from a precomputed database of structural neighbors (see Methods section). The times reported for 3dSEARCH do not include the time required to construct the hash table of secondary structure vectors (see Methods section) since it is a start-up cost that needs to be incurred only once for a given set of target structures. For the 685 structures in our study, the time required to build the 3dSEARCH hash table was 42 seconds.

Discussion

The results in Figure 1 and Tables 1-6 show that the DALI algorithm performs consistently well on each of the three query structures. In the case of myoglobin and TIM, some of the other methods also do as well as DALI, yielding similar or marginally better sensitivity and specificity. For the case of the immunoglobulin query, the DALI program outperforms all of the other techniques by a somewhat larger margin. For example, at a sensitivity of 92% DALI reports 3 false positives while the next best technique LOCK, reports 11 false positives. The immunoglobulin query poses a more significant challenge to structural alignment programs because of the more subtle structural similarities between proteins that share its sandwiched β-sheet fold. For instance, a small change in the relative distance or orientation of each pair of adjacent β-strands tends to have a large cumulative effect on this structure while still maintaining the same sandwich-like fold. We believe that the superior performance of the DALI algorithm for this structure is due to the elastic similarity score that this method uses (see Methods section). Since DALI is based on comparing intra-molecular distances, the effect of the cumulative deviations between pairs of β-strands is relatively low. In addition, since DALI attempts to find the globally optimal arrangement of locally similar regions using the branch-and-bound algorithm, the search space covered by this technique is larger than that covered by most of the other programs in this study. This robustness of the DALI search algorithm is reflected in its increased sensitivity and specificity as well its significantly higher execution times (as compared to LOCK).

LOCK performs as well or better than all of the other techniques for the myoglobin and TIM query structures. In the case of TIM, LOCK does marginally better than DALI, finding 5 false positives at a sensitivity of 98% while DALI finds 10 false positives at the same level. For the immunoglobulin query, LOCK ranks below DALI but above all other structural alignment programs in our test set. One of the primary reasons why LOCK does not perform as well as DALI in this case is the rigid-body constraint that LOCK imposes on its alignments. Since LOCK uses inter-molecular distances to compute the final alignment, the program occasionally has difficulty aligning some structures with substantial cumulative displacements from the query structure. DALI, on the other hand, is able to tolerate some of these cumulative deviations because of its elastic similarity score, which is based on comparing intra-molecular distances. Because of LOCK's rigid-body constraint, one or more β-strands are excluded from the final alignment of some of the distantly related true positive structures, which therefore results in a lower

number of aligned core residues for these structures. Hence, it is only for a few distantly related structures, with either large cumulative displacements within the shared motifs or differently ordered secondary structure elements, that LOCK is not able to find a good alignment. In terms of speed, LOCK is significantly faster than all the other programs that compute structural alignments at the amino-acid level and for which we were able to obtain execution times (i.e. all programs other than 3dSEARCH, which only computes secondary structural alignments, and VAST for which we could not measure execution times). For instance, the execution times required by LOCK are about an order of magnitude lower than the times required by DALI. Hence, LOCK is able to significantly decrease the time required to perform structural comparisons at the cost of a relatively small decrease in sensitivity for certain non-rigid-body alignments.

STRUCTAL also performs well on the myoglobin and TIM query structures, with sensitivity and specificity values only slightly lower than LOCK and DALI. But for the case of immunoglobulin, STRUCTAL reports 159 false positives at a sensitivity of 92%, which is significantly higher than the number of false positives reported by DALI and LOCK at this level (see Table 3). We believe this drop in performance for the immunoglobulin structure is due to the inadequate coverage of the alignment space by this program. STRUCTAL uses multiple starting configurations to improve the likelihood that the iterative dynamic programming technique will locate a global rather than local minimum in alignment space (see Methods section). The six initial alignments that the program uses are probably not adequate for the case of immunoglobulin because of the greater variability in this fold and due to the fact that this fold often appears as a domain within a larger structure. The search space for local alignment of complete chains is significantly larger than the search space for global alignment of smaller structural domains. The STRUCTAL algorithm and scoring scheme (Levitt and Gerstein, 1998) are optimized for the global alignment of domains rather than the local alignment of complete chains. The program therefore has difficulty identifying many of the immunoglobulin true positives that require local rather than global alignment. We verified this claim by converting the 685 target structures into 949 *scop* domains and repeating the immunoglobulin search with this new target set. As expected, STRUCTAL's performance improved significantly (only 13 false positives at a sensitivity of 100%), surpassing that of DALI and LOCK. STRUCTAL's difficulty with the immunoglobulin fold has been noted previously in Gerstein and Levitt, 1998.

The VAST program yields high sensitivity and specificity for the myoglobin and TIM structures but has considerable difficulty with the immunoglobulin query. For this case, VAST finds only 27 of the 38 true positives (i.e., a maximum sensitivity of 71%). VAST's abrupt plateau after 27 true positives is due to the small size of the immunoglobulin query structure (the constant domain of 8fab-A), which contains only 7 β-strands arranged in a 3-4 sandwich. Since VAST is based on matching pairs of secondary structure elements, the probability of finding complete alignments between distantly related structures decreases with the number of secondary structure elements in the query. We verified this hypothesis by repeating the VAST search using the *variable* domain of the immunoglobulin structure (residues 1-105 of 8fab-A) which contains 10 β-strands arranged in a 5-5 sandwich. In this case the program returned 34 of the 38 true positives. It should also be noted that while the target structures for all other programs consisted of complete chains, the targets for the VAST alignments consisted of domains generated by the VAST domain parser. This is due to the fact that we obtained the VAST results from the VAST server, which contains a precomputed database of domain alignments. As described in the previous paragraph, using domains rather than complete chains simplifies the alignment problem by significantly reducing the search space. In addition, since VAST uses a significance score based on the size of the two structures, using complete chains instead of domains could also decrease the performance of the algorithm by lowering the significance of many of the alignments below the threshold of 0.05 (see Methods section). The effect of using complete chains instead of domains can be seen in the case of two of the immunoglobulin true positives (1cdy and 1hng-A) that were missed by VAST because its domain parser had not divided these structures into their constituent domains. Even though the alignments were probably correct, they were considered insignificant because they matched only about half of the secondary structure elements in the target structures.

3dSEARCH performed surprisingly well given the simplicity of its technique and the very low resolution at which it computes its alignments. This program aligns only secondary structure vectors and does not consider correspondences between individual residues while computing the alignment (see Methods section). For both myoglobin and TIM, 3dSEARCH does as well as LOCK and DALI. On the other hand, the sensitivity of this method is relatively low for the immunoglobulin query. The poor performance of 3dSEARCH for the immunoglobulin structure is due to the cumulative effect of variations in the orientation and displacement of the β-strands in this fold, which makes it difficult to perform alignments based only on approximate comparison of

secondary structure vectors. In addition, since 3dSEARCH is based largely on comparing the relative orientation of secondary structure vectors, structures with only β-sheets can be difficult to align since many pairs of β-strands in a sheet can have similar relative orientations (either parallel or anti-parallel). Also, since the scores computed by 3dSEARCH are based only on the number of aligned vectors, the specificity of the program drops more rapidly when the number of vectors in the query structure is small. The principle advantage of 3dSEARCH, as compared to all other alignment programs, is its speed, which is almost two orders of magnitude greater than the next fastest method (LOCK).

Though the MINAREA program yields good results for the myoglobin query, it shows relatively low sensitivity and specificity for the TIM and immunoglobulin query structures. MINAREA's poor performance is primarily due to the algorithm it uses to find the initial alignment that seeds the area-minimization step (see Methods section). This initial alignment is found by simply sliding the smaller structure along the larger structure and selecting the alignment that results in the smallest root mean squared deviation (RMSD). This process does not adequately search the alignment space and is hence likely to miss the global minimum. The probability of finding a good initial alignment is specially low when two structures are of similar size or when one of the structures contain large insertions or deletions. In addition, MINAREA has difficulty aligning multi-domain structures because the algorithm often includes unaligned domains as a large loop region in the final score. Since we executed the MINAREA runs with default parameters, it is possible that adjusting certain parameters could alleviate some of these difficulties.

Acknowledgements

This work was supported by the NLM R01 LM05716-01 grant. The authors thank Steve Brenner, Steve Bryant, Fred Cohen, Mark Gerstein, Liisa Holm, and Werner Krebs for their assistance in obtaining the programs and data for this study and for discussions and critiques of this paper.

References

Artymiuk, P.J., Poirrette, A.R., Rice, D.W., and Willett, P. (1997) A polymerase I palm in adenylyl cyclase? *Nature,* **388**, 33-34.

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.

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.

Brenner, S.E., Chothia, C., and Hubbard, T.J. (1997) Population statistics of protein structures: lessons from structural classifications. *Curr. Opin. Struct. Biol.,* **7**, 369-376.

Bryant, S.H., Madej, T., Janin, J., Liu, Y., Ruoho, A., Zhang, G., and Hurley J.H. (1997) A polymerase I palm in adenylyl cyclase? A reply. *Nature,* **388**, 34.

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

Cohen, G.H. (1997) ALIGN: A program to superimpose protein coordinates, accounting for insertions and deletions. *J. Appl. Crystallogr.*, In press.

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 pair-wise and multiple alignments of protein structures. In *Proc. Fourth Int. Conf. on Intell. Sys. for Mol. Biol.* Menlo Park, CA: AAAI Press. pp 59-67.

Gerstein, M and Levitt, M. (1998) Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Sci.*, **7**, 445-456.

Gibrat, J.F., Madel, T., and Bryant, S.H. (1996) Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.,* **6**, 377-385.

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

Grindley, H.M., Artymuik, P.J., Rice, D.W., and Willett, P. (1993) Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *J. Mol. Biol.* **229**, 707-721.

Hobohm, W, Scharf, M., Schneider, R., and Sander, C. (1992) Selection of representative protein data sets. *Protein Sci.*, **1**, 409-417.

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

Holm, L. and Sander, C. (1996b) Alignment of three-dimensional protein structures: network server for database searching. *Methods Enzymol.,* **266**, 653-662.

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

Holm, L and Sander, C. (1995) 3-D Lookup: Fast Protein Structure Database Searches at 90% Reliability. . In *Proc. Third Int. Conf. on Intell. Sys. for Mol. Biol.* Menlo Park, CA: AAAI Press. pp 179-187.

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

Holm, L. and Sander, C. (1993b) Structural alignment of globins, phycocyanins, and colicin. *FEBS Lett.,* **315**, 301-306.

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

Levitt, M. and Gerstein, M. (1998) A unified statistical framework for sequence comparison and structure comparison. *PNAS,* **95**, 5913-5920.

Lamdan, Y. and Wolfson, H.J. (1988) Geometric Hashing: A general and efficient model based recognition scheme. . In *Proc. IEEE Int. Conf. on Computer Vision*. pp 238-249.

Lathrop, R.H. (1994) The protein threading problem with sequence amino acid interaction preferences in NP-complete. *Protein Eng.,* **7**, 1059-1068.

Lathrop, R.H. and Smith, T.F. (1996) Global optimal protein threading with gapped alignment and empirical pair potentials. *J. Mol. Biol.,* **255**, 641-665.

Madej, T., Gibrat, J.F., and Bryant, S.H. (1995) Threading a database of protein cores. *Proteins,* **23**, 356-369.

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.

Murzin, A., Brenner, S.E., Hubbard, T., and Chothia, C. (1995) *scop*: A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.,* **247**, 536-540.

Needleman S.B. and Wunsh, C.D.. (1971) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48**, 443-453.

Orengo, C.A. and Taylor, W.R. (1996) SSAP: Sequential Structure Alignment Program for protein structure comparison. *Methods Enzymol.,* **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,* **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.

Satow, Y., Cohen, G.H., Padlan, E.A., and Davies, D.R. (1987) Phosphocholine binding immunoglobulin Fab McPC603: An X-ray diffraction study at 2.7 Å. *J. Mol. Biol.,* **190**, 593-604.

Schuler, G.D., Epstein, J.A., Ohkawa, H., and Kans, J.A. (1996) Entrez: Molecular biology database and retreival system. *Methods Enzymol.,* **266**, 141-162

Singh, A.P. and Brutlag, D.L., (1997) Hierarchical protein structure superposition using both secondary structure and atomic representations. In *Proc. Fifth Int. Conf. on Intell. Sys. for Mol. Biol*. Menlo Park, CA: AAAI Press. pp 284-293.

Subbiah, S., Laurents, D.V., and Levitt, M. (1993) Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core. *Curr. Biol.,* **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**, 55-78.

Table 1. Number of false positives at various sensitivities for myoglobin (1mbd).

Number of True Positives	Corresponding Sensitivity	DALI	STRUCTAL	VAST	MINAREA	LOCK	3dSEARCH
7	63.6%	0	0	0	0	0	0
8	72.7%	0	1	0	0	0	1
9	81.8%	0	1	0	0	0	1
10	90.9%	0	1	1	0	0	2
11	100%	0	1	2	0	0	3

Table 2. Number of false positives at various sensitivities for TIM (1tph-2).

Number of True Positives	Corresponding Sensitivity	DALI	STRUCTAL	VAST	MINAREA	LOCK	3dSEARCH
25	49.0%	0	0	0	2	0	0
30	58.8%	0	1	0	8	0	0
35	68.6%	0	1	0	90	0	0
40	78.4%	0	1	0	161	0	0
45	88.2%	1	1	0	330	1	0
50	98.0%	10	4	---	605	5	4
51	100%	---	163	---	634	24	139

Table 3. Number of false positives at various sensitivities for immunoglobulin (8fab-A).

Number of True Positives	Corresponding Sensitivity	DALI	STRUCTAL	VAST	MINAREA	LOCK	3dSEARCH
10	26.3%	0	0	0	3	0	3
15	39.5%	0	0	0	4	0	6
20	52.6%	0	1	1	17	0	20
25	65.7%	0	3	1	107	1	50
30	78.9%	1	9	---	298	2	66
35	92.1%	3	159	---	573	11	185
38	100%	---	425	---	642	383	322

Table 4. True positives (+) and false negatives (-) for myoglobin at a specificity cutoff of 90% (1 false positive).

PDB ID	Protein Description	DALI	STRUC-TAL	VAST	MIN-AREA	LOCK	3dSEA-RCH
1mbd	Myoglobin	+	+	+	+	+	+
1bab-B	Hemoglobin, beta-chain	+	+	+	+	+	+
2fal	Myoglobin	+	+	+	+	+	+
1eca	Erythrocruorin	+	+	+	+	+	+
2hbg	Glycera globin	+	+	+	+	+	+
1hlb	Hemoglobin	+	+	+	+	+	+
1ash	Ascarishemoglobin, domain 1	+	+	+	+	+	+
3sdh-A	Hemoglobin I	+	+	+	+	+	+
2gdm	Leghemoglobin	+	+	+	+	+	+
1cpc-A	C-phycocyanin	+	+	+	+	+	-
1cpc-B	C-phycocyanin	+	+	-	+	+	-

Table 5. True positives (+) and false negatives (-) for TIM at a specificity cutoff of 90% (5 false positives).

PDB ID	Protein Description	DALI	STRUC-TAL	VAST	MIN-AREA	LOCK	3dSEA-RCH
1tph-2	Triosephosphate isomerase	+	+	+	+	+	+
5tim-A	Triosephosphate isomerase	+	+	+	+	+	+
1nsj	N-(5'phosphoribosyl)antranilate (PRA)isomerase	+	+	+	+	+	+
1fba-A	Fructose-1,6-bisphosphate aldolase	+	+	+	+	+	+
1jul	Indole-3-glycerophosphate (IGP) synthase	+	+	+	-	+	+
1dhp-A	Dihidrodipicolinate synthase	+	+	+	+	+	+
2tys-A	Trp synthase alpha-subunit	+	+	+	+	+	+
2tmd-A	Trimethylamine dehydrogenase, N-term domain	+	+	+	-	+	+
1nal-1	N-acetylneuraminate lyase	+	+	+	+	+	+
1ltd-A	Flavocytochrome b2, C-terminal domain	+	+	+	-	+	+
1oyc	Old yellow enzyme (OYE)	+	+	+	+	+	+
1nar	Seed storage protein	+	+	+	+	+	+
1pkm	Pyruvate kinase, N-terminal domain	+	+	+	-	+	+
1cnv	Seed storage protein	+	+	+	+	+	+
1ctn	Chitinase A, central domain	+	+	+	+	+	+
1ceo	Endoglucanase CelC	+	+	+	+	+	+
1luc-A	Bacterial luciferase	+	+	+	+	+	+
1dor-A	Dihydroorotate dehydrogenase A	+	+	+	+	+	+
1gow-A	beta-Glycosidase	+	+	+	+	+	+
1ece-A	Endocellulase E1	+	+	+	+	+	+
1dos-A	Fructose-bisphosphate aldolase	+	+	+	+	+	+
2myr	Myrosinase	+	+	+	+	+	+
1qba	Bacterial chitobiase, catalytic domain	+	+	+	+	+	+
1pud	tRNA-guanine transglycosylase	+	+	+	+	+	+
1ghr	plant beta-glucanases	+	+	+	+	+	+
2acq	Aldose reductase	+	+	+	+	+	+
1sft-A	Alanine racemase, N-terminal domain	+	+	+	+	+	+
1req-A	Methylmalonyl-CoA mutase, alpha & beta subunits	+	+	+	+	+	+
1byb	beta-Amylase	+	+	+	+	+	+
2ebn	Endo-beta-N-acetylglucosaminidase	+	+	+	+	+	+
2mnr	Mandelate racemase	+	+	+	+	+	+
1edg	Endoglucanase CelA	+	+	+	+	+	+
5rub-A	Ribulose 1,5-bisphosphate carboxylase-oxygenase	+	+	+	+	+	+
2amg	G4-amylase (1,4-alpha-D-glucan maltotetrahydrolase)	+	+	+	+	+	+
1psc-A	Phosphotriesterase	+	+	+	+	+	+
1ucw-A	Transaldolase	+	+	+	+	+	+
1tcm-A	Cyclodextrin glycosyltransferase	+	+	+	+	+	+
1bgl-A	beta-Galactosidase, domain 3	+	+	+	+	+	+
1xyz-A	Xylanase XynZ (family F)	+	+	+	+	+	+
2chr	Chlormuconate cycloisomerase	+	+	+	+	+	+
1qap-A	quinolinic acid phosphoribosyltransferase	+	+	+	+	+	+
2aaa	Fungal alpha-amylases	+	+	+	+	+	+
1fkx	Adenosine deaminase	+	+	+	+	+	+
2kau-C	alpha-subunit of urease, catalytic domain	+	+	+	+	+	+
4xia-A	D-xylose isomerase	+	+	+	+	+	+
4enl	Enolase	+	+	+	+	+	+
1smd	Mammalian alpha-amylase	+	+	+	+	+	+
1gym	Phosphatidylinositol-specific phospholipase C	+	+	+	+	+	+
1djx-A	Phospholipase C isozyme D1 (PLC-D1)	+	+	+	+	+	+
1nfp	non-fluorescent flavoprotein (luxF, FP390)	-	+	-	+	+	+
1bpl-A	Bacterial alpha-amylase (BLA)	-	-	-	+	-	-

Table 6. True positives (+) and false negatives (-) for immunoglobulin at a specificity cutoff of 90% (4 false positives).

PDB ID	Protein Description	DALI	STRUC-TAL	VAST	MIN-AREA	LOCK	3dSEA-RCH
8fab-A	Immunoglobulin	+	+	+	+	+	+
1osp-L	Immunoglobulin	+	+	+	+	+	+
1fc2-D	Immunoglobulin	+	+	+	+	+	+
1dlh-B	Class II MHC	+	+	+	+	+	+
1dlh-A	Class II MHC	+	+	+	+	+	-
8fab-B	Immunoglobulin	+	+	+	+	+	+
1mhc-A	Class I MHC, beta2-microglobulin and alpha-3 domain	+	+	+	-	+	-
1bec	T-cell antigen receptor	+	-	+	+	+	+
1fru-A	Fc (IgG) receptor, alpha-3 domain and beta subunit	+	+	+	-	+	+
2ncm	Neural cell adhesion molecule (NCAM)	+	+	+	+	+	-
1tlk	Telokin	+	+	+	+	+	-
1zxq	Intercellular cell adhesion molecule-2 (ICAM-2)	+	+	+	+	+	-
1vsc-A	Vascular cell adhesion molecule-1 (VCAM-1)	+	+	+	+	+	-
1tcr-A	T-cell antigen receptor	+	+	+	-	-	-
1neu	Myelin membrane adhesion molecule P0	+	+	+	-	+	-
1hng-A	CD2	+	-	-	-	+	+
1fnf	Fibronectin, fifferent Fn3 modules	+	+	+	-	+	+
1ebp-A	Erythropoietin (EPO) receptor	+	+	+	-	+	-
1bgl-A	beta-Galactosidase, domains 2 and 4	+	+	-	-	+	-
1cid	CD4	+	-	+	-	+	-
1fie-A	Transglutaminase factor XIII, N-terminal domain	+	+	+	-	+	+
4kbp-A	Purple acid phosphatase, N-terminal domain	+	-	+	-	+	-
3hhr-C	Growth hormone receptor	+	-	+	+	+	-
3dpa	Pilus chaperone PapD, N-domain	+	+	-	-	+	-
2hft	Extracellular region of human tissue factor	+	+	+	+	+	-
1svb	Envelope glycoprotein, domain III (C-terminal)	+	-	-	-	-	-
1ctn	Chitinase A, N-terminal domain	+	-	-	-	-	+
1cfb	Neuroglian, the two amino proximal Fn3 repeats	+	+	+	+	+	-
1cdy	CD4	+	+	-	-	+	-
1clc	CelD cellulase, N-terminal domain	+	+	+	-	+	-
1oxy	Hemocyanin, C-terminal domain	+	-	-	-	-	-
1jcv	Cu,Zn superoxide dismutase, SOD	+	+	-	-	+	+
1gof	Galactose oxidase, C-terminal domain	+	+	-	-	-	+
1nfk-A	p50 subunit of NF-kappa B transcription factor	+	+	+	-	+	-
1msp-A	Major sperm protein, alpha isoform (recombinant)	+	+	+	+	+	-
1tcm-A	Cyclodextrin glycosyltransferase, domain E	+	-	-	-	-	-
1edh-A	E-Cadherin domains 1 and 2	+	+	+	+	+	-
1qba	Bacterial chitobiase, c-terminal domain	-	-	-	-	-	-

Table 7. Execution times (hrs:min:sec) for aligning each query structure to 685 target structures.

Query	Number of Residues	DALI	STRUCTAL	MINAREA	LOCK	3dSEARCH
1mbd	153	1:0:0	2:38:55	0:57:09	0:06:28	0:0:04
1tph-2	245	4:0:0	3:18:05	1:17:07	0:13:05	0:0:27
8fab-A	103	1:0:0	1:50:16	0:45:44	0:06:54	0:0:07

Figure Legend

Figure 1. ROC curves for the six structural alignment programs.