2-5-2010

Fast Protein Structure Alignment

Yosi Shibberu
Rose-Hulman Institute of Technology, shibberu@rose-hulman.edu

Allen Holder
Rose-Hulman Institute of Technology, holder@rose-hulman.edu

Kyla Lutz
Rose-Hulman Institute of Technology, kyla.lutz@rose-hulman.edu

MSTR 10-01

Fast Protein Structure Alignment

Yosi Shibberu, Allen Holder, and Kyla Lutz

Mathematical Sciences Technical Report Series
MSTR 10-01

February 5, 2010

Department of Mathematics
Rose-Hulman Institute of Technology
http://www.rose-hulman.edu/math

Fax (812)-877-8333 **Phone (812)-877-8193**

Fast Protein Structure Alignment

Yosi Shibberu[1], Allen Holder[2], and Kyla Lutz[3]

[1] Rose-Hulman Institute of Technology, Department of Mathematics,
Shibberu@rose-hulman.edu
[2] Rose-Hulman Institute of Technology, Department of Mathematics
Holder@rose-hulman.edu
[3] Rose-Hulman Institute of Technology, Department of Mathematics
Kyla.Lutz@rose-hulman.edu

Abstract. We address the problem of aligning the 3D structures of two proteins. Our pairwise comparisons are based on a new optimization model that is succinctly expressed in terms of linear transformations and highlights the problem's intrinsic geometry. The optimization problem is approximately solved with a new polynomial time algorithm. The worst-case analysis of the algorithm shows that the solution is bounded by a constant depending only on the data of the problem.

1 Introduction and Background

Proteins play a key role in nearly all biochemical processes of a living organism. The three dimensional structure of a protein molecule largely determines its biological function, and inferences can be made about one protein's function by aligning it to others whose biological function is already established [21]. Hence, protein structure alignment is an important problem in biology.

A protein is a long chain assembled from twenty different types of amino acids called residues. Protein chains fold into unique, tightly packed, globular structures called folds. Typically, a protein's fold is specified by a list of the three dimensional coordinates of each atom in the protein. A distance matrix specifying all the distances between pairs of atoms in the protein completely determines the fold up to reflections in a coordinate invariant way [12]. A distance matrix is often converted into a contact matrix, or map, whose entries equal one for pairs of atoms within a certain cut-off distance from one another and zero otherwise.

The objective in protein alignment is to determine a one-to-one correspondence between a subset of the atoms or residues in two different protein structures. The subset chosen should optimize some biologically relevant similarity measure, although there is currently no consensus on what this measure of similarity should be [21]. In fact, the structure alignment problem itself may not be well-posed in all cases [10].

Existing protein alignment algorithms largely fall into two categories: (i) algorithms that directly use the three dimensional Cartesian coordinates of the atoms and (ii) algorithms that use internal coordinates (e.g. contact matrices) as a basis for comparisons [21]. Unlike sequence alignment, exact polynomial-time

structure alignment algorithms do not exist. Kolodny and Linial [18] claim it is possible to obtain an approximate polynomial-time algorithm if one exploits three-dimensional Euclidean geometry. Their claim seems to favor alignment algorithms from category (i). However, three dimensional Euclidean geometry based alignments may introduce undesirable rigidity in the alignment problem [20]. Contact matrix based alignments may be more biologically relevant since they increase flexibility.

The contact map overlap (CMO) protein alignment problem is the problem of determining a one-to-one correspondence between subsets of residues in two proteins that maximizes the overlap of their contact matrices [1, 2, 6, 8, 19, 22, 25]. The CMO problem can be shown to be equivalent to other, well-studied optimization problems, like the maximum subgraph problem [1, 2], and is known to be NP-complete [11].

Integer programming formulations of the CMO problem have been solved with branch-and-bound techniques and several associated relaxations [2, 6, 8, 19, 25]. The problem was originally formulated in [19] as a binary, quadratic problem. Relaxations of this formulation are studied in [2, 8] and an exact algorithm was developed in [25]. A fast CMO algorithm that exploits a special structure of the maximum clique problem is described in [22], and a technique that leverages the special properties of self-avoiding walks in two and three-dimensional Euclidean space is developed in [1].

Our approach to protein structure alignment is different. First, we do not use discrete contact maps but instead smooth the contact information and reformulate the problem in n-dimensional Euclidean space, see Figure 1. Second, our geometric reformulation bounds our optimization problem by constructing a solution to the underlying combinatorial problem. Third, integer programming formulations attempt to align proteins using local contact information. We instead take a global perspective by first decomposing the contact maps and identify a smaller collection of characteristic subspaces on which to make alignments. Our method competes favorably with other recently published methods for the CMO problem in terms of time and quality, and our algorithm should scale well with problem size.

2 Notation and Problem Statement

Let X be the $n \times 3$ coordinate matrix whose ith row is the coordinates of the ith atom, and let M be the $n \times n$ *distance matrix* whose (i, j) element is the distance between atom i and atom j, i.e.

$$M_{i,j} = \|X_{i,:} - X_{j,:}\|,$$

where $X_{i,:}$ and $X_{j,:}$ are the i-th and j-th columns of X. The matrices X and M are known to be in a one-to-one relationship up to reflection [12].

We let

$$[C(\rho, \kappa)]_{ij} = \max\{\min\{-\rho(M_{i,j} - \kappa), 1\}, 0\},$$

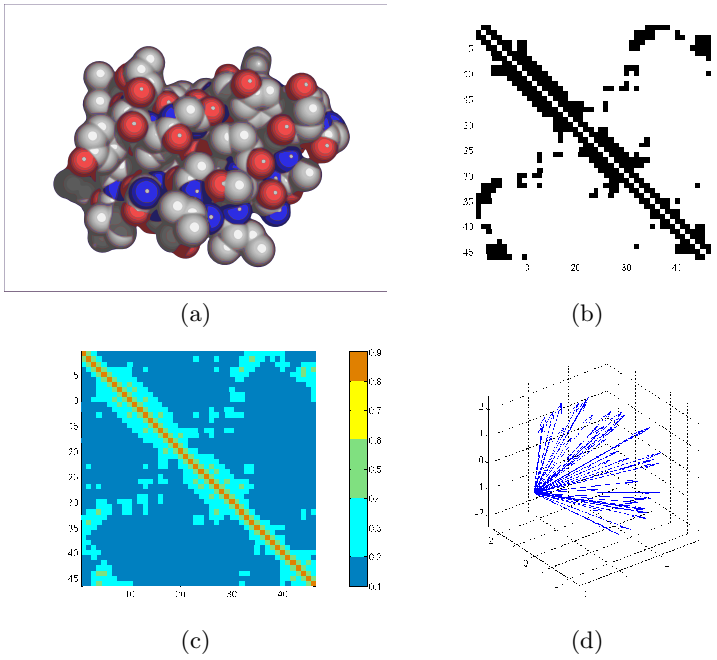


Fig. 1. Representations of the fold of the protein crambin (1crn). (a) 3D representation (b) 8A contact map (c) smoothed positive-definite contact map (d) intrinsic contact vectors projected to R^3.

which is a smooth contact matrix, see Figure 2 for a graph of the piecewise linear function. The parameter κ is the distance cutoff parameter and ρ is the magnitude of the slope of the sigmoid. Importantly, if $\rho = 1/\kappa$, then $[C(\rho,\kappa)]_{i,j}$ is arbitrarily small for $i \neq j$ as κ decreases to zero. Hence, we can ensure $C(\rho,\kappa)$ is diagonally dominant and subsequently positive definite. We make this assumption throughout.

Let $C'(\rho,\kappa) = C'$ and $C''(\rho,\kappa) = C''$ be contact matrices for two different proteins for which we assume, at least for now, that the number of residues is the same. Although this assumption is atypical, this allows us to succinctly study the fundamentals of our alignment problem, and importantly, it highlights the combinatorial difficulty that we overcome. We adapt our study to the more realistic case of the two proteins having a different number of residues in Section 4.

The assumption that both ρ and κ are selected so that both C' and C'' are positive definite means that there are unitary matrices U and W so that

$$C' = UD'U^T \text{ and } C'' = WD''W^T,$$

where D' and D'' are the diagonal matrices comprised of the positive eigenvalues for C' and C''. Since the eigenspaces and eigenvalues characterize the contact matrices, it makes sense to align them. The essence of our comparison technique

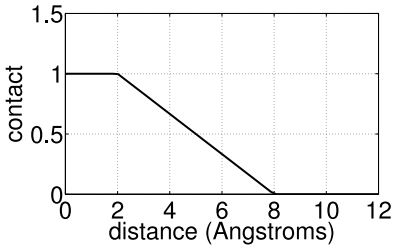


Fig. 2. The graph of the function $\max\{\min\{-\rho(M_{i,j}-\kappa),1\},0\}$ for $\rho = 1/6$ and $\kappa = 8$. The horizontal axis is $M_{i,j}$ Angstroms.

rests on the fact that the orthonormality of U and W ensures that we can find a rotation matrix Θ that perfectly aligns U with W, i.e. we can guarantee $\Theta W = U$. However, we have a different rotation for each of the 2^n orientations of the eigenvectors. For example, if we replace the first column of U with its negative, then we have a different rotation. Deciding an optimal rotation means addressing the possibility of searching through all 2^n possible orientations.

Three collections of linear operators define our search space, and we let

- $\mathcal{P}$ be the collection of all permutation matrices,
- $\mathcal{R}$ be the collection of all rotation matrices, and
- $\mathcal{I}$ be the collection of all axial reflections, i.e. $\mathcal{I}$ is the set of diagonal matrices $I^{\pm}$ for which each diagonal element is either 1 or -1.

The alignment problem we propose is

$$\min\left\{\|C' - \Theta C''\Omega\|_p^p : \Theta W = UI^{\pm},\ I^{\pm} \in \mathcal{I},\ \Theta \in \mathcal{R},\ \Omega \in \mathcal{P}\right\}. \tag{1}$$

The matrix $I^{\pm}$ orients the eigenvectors of C', for which the unique rotation $\Theta = UI^{\pm}W^T$ aligns the eigenvectors of C'' with those of C'. The permutation matrix Ω pairs the contact vectors to minimize the deviation as measured by the matrix p-norm (we assume throughout that $1 \leq p \leq 2$ so that the sub-multiplicative property holds). We mention that the extreme case in which $\rho \to \infty$ places the problem in graph theoretical terms since both C' and C'' are adjacency matrices for a graph (V, E), with V being the set of respective residues $\{r_1, r_2, \ldots, r_n\}$ and $E = \{(r_i, r_j) : D_{ij} < \kappa\}$.

The problem can be re-written since the constraint $\Theta = UI^{\pm}W^T$ gives

$$C' - \Theta C''\Omega = UD'U^T - UI^{\pm}W^TWD''W^T\Omega = U(D'U^T - I^{\pm}D''W^T\Omega).$$

Using the sub-multiplicative property, we can re-state the problem as

$$\min\left\{\|D'U^T - I^{\pm}D''W^T\Omega\|_p^p : I^{\pm} \in \mathcal{I},\ \Omega \in \mathcal{P}\right\}. \tag{2}$$

Moreover, for the 2-norm we have

$$\begin{aligned}
&\|D'U^T - I^{\pm}D''W^T\Omega\|_2^2 \\
&= \operatorname{tr}\left(\left(D'U^T - I^{\pm}D''W^T\Omega\right)^T\left(D'U^T - I^{\pm}D''W^T\Omega\right)\right) \\
&= \operatorname{tr}\left((U(D')^2U^T - 2UD'I^{\pm}D''W^T\Omega - \Omega^TW(D')^2W^T\Omega\right),
\end{aligned}$$

where $\operatorname{tr}(\cdot)$ is the trace of the matrix. Both $U(D')^2U^T$ and $\Omega^TW(D'')^2W^T\Omega$ are constants under the trace calculation, which means that a 2-norm reformulation is

$$\max\left\{\operatorname{tr}\left(UD'I^{\pm}D''W^T\Omega\right) : I^{\pm} \in \mathcal{I},\ \Omega \in \mathcal{P}\right\}. \tag{3}$$

The 2-norm formulation along with the positive definite assumption provides an intrinsic geometric description of the similarity measure we optimizing. Let $R' = \sqrt{D'}U^T$, where the square root is elementwise. We refer to the columns of R' as the *intrinsic contact vectors* of a protein, and each of these corresponds to a residue. Recall that $C'_{i,j}$ is the contact between residue i and residue j. Since $C' = (R')^TR'$, we have that $C'_{i,j} = (R')^T_{:,i}R'_{:,j}$. Moreover, since the diagonal elements of C' are equal to one (every residue is in contact with itself), we have that the intrinsic contact vectors are unit vectors since $C'_{ii} = (R')^T_{:,i}R'_{:,i} = 1$. Therefore, we can interpret the contact between two residues of a protein as the cosine of the angle between their corresponding intrinsic contact vectors. Allowing $R'' = \sqrt{D''}W^T$, we see that

$$D'U^T = \sqrt{D'}R' \ \text{ and } D''W^T = \sqrt{D''}R'',$$

which means the objective function in (3) is

$$\operatorname{tr}\left(R'\sqrt{D'}I^{\pm}\sqrt{D''}R''\Omega\right).$$

This shows that the 2-norm objective is a scaled sum of the cosines of the angles between the paired intrinsic contact vectors from the two proteins. Since the maximum value of the cosine is 1 if the angle is zero, we have the geometric insight that the 2-norm objective is minimizing the angles between the paired intrinsic contact vectors.

3 Algorithmic Motivation

The optimization problem in (2) can be modeled as a mixed integer optimization problem (MIP), for which a number of exact algorithms are known. However, the binary search tree underlying the MIP formulation has 2^n leaves, each of which corresponds to a unique $I^{\pm}$ in $\mathcal{I}$. For any one of these an optimal permutation matrix Ω can be calculated by solving a traditional assignment problem on the bipartite graph (N', N'', E), where N' is the collection of column vectors in $D'U^T$, N'' is the collection of column vectors in $I^{\pm}D''W^T$, $E = N' \times N''$, and each edge is weighted with the p-norm difference of the defining vectors. While the assignment problem is polynomial, the fact that we might have to solve 2^n

of these problems is cause for concern since n is typically around a 100. To test the ability of stock solvers we formed the MIP in AMPL and tried to solve a 10 residue problem with MINLP (posted at NEOS, `http://www-neos.mcs.anl.gov/`). The solution was known to be $I^{\pm} = \Omega = I$. However, MINLP reported a different optimal solution with an objective value of about 10 times that of the known optimum. As a counterpart, CPLEX correctly identified the solution by solving the standard MIP relaxation. Unfortunately, similar success for larger, and more difficult, problems was not observed with CPLEX. This demonstrates the need for quick, high-quality heuristics to align large proteins, and we present a new, polynomial-time search strategy that is based on a geometric bound.

A small example highlights that the assignment problem is bounded for each $I^{\pm}$. The following are from 3 atoms of a beta sheet in two different proteins,

$$D'U^T = \begin{bmatrix} 0.0066 & -0.0128 & 0.0066 \\ 0.0953 & -0.0002 & -0.0950 \\ 1.6278 & 1.6793 & 1.6281 \end{bmatrix}$$

and

$$D''W^T = \begin{bmatrix} 0.0036 & -0.0070 & 0.0036 \\ 0.0104 & -0.0002 & -0.0103 \\ 1.6223 & 1.6819 & 1.6225 \end{bmatrix}.$$

We construct $I^{\pm}$ by minimizing the maximum magnitude of each row of $D'U^T - I^{\pm}D''W^T\Omega$. For example, if the first diagonal element of $I^{\pm}$ is 1, then the maximum magnitude element of the first row of $D'U^T - I^{\pm}D''W^T\Omega$ is

$$\begin{aligned} 0.0194 = &\max\{0.0066, -0.0128, 0.0066, 0.0036, -0.0070, 0.0036\} \\ &- \min\{0.0066, -0.0128, 0.0066, 0.0036, -0.0070, 0.0036\}. \end{aligned}$$

If the first diagonal element of $I^{\pm}$ is instead -1, the maximum magnitude element of the first row of $D'U^T - I^{\pm}D''W^T\Omega$ is

$$\begin{aligned} 0.0198 = &\max\{0.0066, -0.0128, 0.0066, -0.0036, 0.0070, -0.0036\} \\ &- \min\{0.0066, -0.0128, 0.0066, -0.0036, 0.0070, -0.0036\}. \end{aligned}$$

Since the first is lower, we let the first diagonal element of $I^{\pm}$ be 1. For the second diagonal element we find that the maximum possible magnitude difference in the second row is 0.2071 if we choose either 1 or -1, which leaves this element undecided. For the third diagonal element we have a maximum possible magnitude difference of 0.0596 for 1 and 3.3612 for -1, and we select the 1 over the -1. This leaves two choices for the diagonal elements of $I^{\pm}$, either $(1, 1, 1)$ or $(1, -1, 1)$.

This construction of $I^{\pm}$ guarantees the magnitude of the difference between each matrix coefficient of $D'U^T - I^{\pm}D''W^T\Omega$ is at most the corresponding row value independent of Ω. So, for either of our two choices of $I^{\pm}$ we have for any permutation matrix Ω that

$$\left|D'U^T - I^{\pm}D''W^T\Omega\right| \leq \begin{bmatrix} 0.0194 & 0.0194 & 0.0194 \\ 0.2071 & 0.2071 & 0.2071 \\ 0.0056 & 0.0056 & 0.0056 \end{bmatrix},$$

where the absolute value of the matrix is componentwise. This bounds the optimal value of (2) by $3\|(0.0194, 0.2071, 0.0056)^T\|_p^p$, which for $p = 2$ is 0.1299. This problem's unique optimal solution has both $I^{\pm}$ and Ω being the identity, which gives an optimal value 0.0001. So the technique identified the optimal $I^{\pm}$. Importantly, the technique also identified the two $I^{\pm}$ matrices with the lowest objective values, which are listed in Table 1 for all $I^{\pm}$ and Ω possibilities. The calculation identifying the third diagonal element of $I^{\pm}$ hints that there is possibly a relatively large assignment if -1 is selected. Table 1 shows that the best assignment if the third diagonal is -1 is $O(10^4)$ above the assignments in which the third diagonal is 1.

$I^{\pm} \setminus \Omega$	(1,2,3)	(1,3,2)	(2,1,3)	(2,3,1)	(3,2,1)	(3,1,2)
(1, 1, 1)	0.0001	0.0263	0.0266	0.0592	0.0790	0.0591
(1, 1, -1)	32.4270	32.4210	32.4210	32.4210	32.4270	32.4210
(1, -1, 1)	0.0790	0.0594	0.0591	0.0264	0.0001	0.0264
(-1, 1, 1)	0.0006	0.0258	0.0265	0.0594	0.0790	0.0594
(1, -1, -1)	32.4270	32.4210	32.4210	32.4210	32.4270	32.4210
(-1, 1, -1)	32.4270	32.4210	32.4210	32.4210	32.4270	32.4210
(-1, -1, 1)	0.0790	0.0595	0.0593	0.0260	0.0006	0.0260
(-1, -1, -1)	32.4270	32.4210	32.4210	32.4210	32.4270	32.4210

Table 1. The first column lists the diagonal elements of $I^{\pm}$, so the diagonal of $I^{\pm}$ for the second row is $(1, 1, -1)$. The first row shows the permutation used to order the columns of the identity to form Ω. So, Ω for the second column has the 2nd and third columns of the identity swapped. For ease of presentation the values are rounded to four decimal places, which leaves two values at 0.0001. However, the top, left most value is lower with increased accuracy.

From a geometric perspective the construction of $I^{\pm}$ orients, or signs, the axial components of the column vectors of $D''W^T$ so that they collapse into the smallest "box" that also contains the column vectors of $D'U^T$. This box bounds the worst possible assignment. Formally, for $\eta_i \in \{1, -1\}$ we let

$$\delta_i^{\min}(\eta_i) = \min_j \left(\{\lambda_i' U_{j,i}\} \cup \{\eta_i \lambda_i'' W_{j,i}\}\right)$$

and

$$\delta_i^{\max}(\eta_i) = \max_j \left(\{\lambda_i' U_{j,i}\} \cup \{\eta_i \lambda_i'' W_{j,i}\}\right).$$

Then setting $\Delta_i(\eta_i) = \left(\delta_i^{\max}(\eta_i) - \delta_i^{\min}(\eta_i)\right)$, we have

$$\max_{\Omega \in \mathcal{P}} \left\{\|D'U^T - I^{\pm} D'' W^T \Omega\|_p^p : I_{i,i}^{\pm} = \bar{\eta_i} \ \forall i\right\} \leq n \sum_i \min\{\Delta_i(1), \Delta_i(-1)\},$$

where $\bar{\eta_i}$ satisfies $\Delta_i(\bar{\eta_i}) = \min\{\Delta_i(1), \Delta_i(-1)\}$. Since the particular $I^{\pm}$ used here is only one of the 2^n elements of $\mathcal{I}$, we have the following

Theorem 1. *The optimal value of the alignment problem in* (1) *is no worse than* $n \sum_i \min\{\Delta_i(1), \Delta_i(-1)\}$.

Since calculating all Δ_is is $O(n^2)$, Theorem 1 gives a polynomial upper bound on the problem. Our experimental results show that this bound is not generally indicative of the optimal value of (1), especially if the proteins align well. This is not surprising since the bound is a worse case estimate of the geometry of the problem, and in the case that the proteins align well, the geometric bound is expected to be a poor estimate of the alignment problem. However, there is significant value in calculating the bound since it identifies meaningful orientations. For example, suppose that $\Delta_i(1) << \Delta_i(-1)$. This suggests a preference to sign the i-th eigenvector with a 1 since if we instead select -1, the column vectors of $I^{\pm}D''W^T$ deviate from the column vectors of $D'U^T$. Since our goal is to minimize deviation, we select 1.

4 Adaptations for Real Numerical Studies

The previous section presents a method of calculating $I^{\pm}$ so that the assignment problem is bounded geometrically, and in this section we develop a polynomial time solution procedure based on this calculation. We first adapt our model to the more realistic case in which

- the number of residues differs between the two proteins, and
- residues from like secondary structures are aligned.

We assume for convenience that the protein with the fewer number of residues corresponds to C'. In this case we pad C' with rows and columns of zeros to the right and to the bottom so that its dimensions agree with C''. Unlike the simplified case studied earlier, part of the alignment problem is to select the eigenvectors of the larger protein that best aligns with the smaller protein. Let there be n_1 residues in the smaller protein and n_2 in the larger. The required selection is accomplished by a linear operator of the form

$$\Gamma = \begin{bmatrix} \Gamma' \\ \cdots \\ 0 \end{bmatrix},$$

in which Γ' is a $n_1 \times n_2$ binary matrix whose row sums are 1. This alters (2) to become

$$\min\left\{\|D'U^T - I^{\pm}\Gamma D''W^T\Omega\|_p^p : I^{\pm} \in \mathcal{I},\ \Omega \in \mathcal{P},\ \Gamma \in \mathcal{G}\right\}, \tag{4}$$

where $\mathcal{G}$ is the collection of all possible Γ matrices. To account for secondary structure alignment we enforce additional restrictions on Ω. Structural motifs, such as β-sheets and α-helices, are identified by the DSSP algorithm due to [15], and part of the alignment problem is to align residues between like structures in the two proteins. Ensuring such alignments is accomplished by altering Ω. In

particular, we assume that $\Omega_{i,j} = 0$ if the secondary structure of residue i in the first protein disagrees with the secondary structure of residue j in the second protein. Since the number of residues in like secondary structures typically varies between the two proteins, we can no longer ensure that that each row and column of Ω contains a single 1, and instead, we can only ensure

$$\sum_i \Omega_{i,j} \leq 1 \text{ for all } j, \ \sum_j \Omega_{i,j} \leq 1 \text{ for all } i, \text{ and } \sum_{i,J} \Omega_{i,j} \leq S. \tag{5}$$

The maximum value of S that the summation in the last condition can achieve is the total number of residues that are in a common secondary structure. For example, if the first protein has 8 residues in an α-helix and 3 residues in a β-sheet, whereas the second protein has 5 residues in an α-helix and 4 in a β-sheet, then the maximum value of S that can be achieved is $\min\{8,5\} + \min\{3,4\} = 8$. Since $\sum_{i,j} \Omega_{i,j}$ is the number of paired residues, we generally want this to be large. If we let $\mathcal{P}'$ be the altered set of Ω matrices, the complete alignment problem we consider is

$$\min\left\{\|D'U^T - I^{\pm}\Gamma D''W^T\Omega\|_p^p : I^{\pm} \in \mathcal{I},\ \Omega \in \mathcal{P}',\ \Gamma \in \mathcal{G}\right\}, \tag{6}$$

which can be re-written in terms of the contact matrices as

$$\min\left\{\|C' - \Theta W \Gamma W^T C''\Omega\|_p^p : \Theta W = UI^{\pm}, I^{\pm} \in \mathcal{I},\ \Omega \in \mathcal{P}',\ \Gamma \in \mathcal{G}\right\}.$$

The only interpretive differences between this and (1) are that $W\Gamma W^T$ projects C'' onto a smaller dimension so that it can be aligned with C' and that $\mathcal{P}'$ is altered from $\mathcal{P}$. As discussed momentarily, both Γ and Ω can be calculated efficiently, which means the combinatorial difficulty remains with calculating $I^{\pm}$. Our algorithmic structure circumvents the combinatorial issue of the problem by calculating the Δ_i's as follows,

1. Calculate Γ with an assignment problem.
2. Use $\Gamma D''$ instead of D'' to calculate Δ_i and let

$$I_{i,i}^{\pm} = \begin{cases} 1, \Delta_i(1) < \Delta_i(-1) \\ -1, \Delta_i(1) > \Delta_i(-1) \\ 0, \Delta_i(1) = \Delta_i(-1). \end{cases}$$

3. Calculate Ω with either an assignment problem or dynamic programming.

The fact that $I_{i,i}^{\pm}$ can be zero means that $I^{\pm}$ is acting as an additional projection, i.e. the product $I^{\pm}\Gamma$ is selecting a collection of eigenvectors as well as signing those that are selected. From the previous example we see that the additional projection identifies the coordinates for which the calculation of Δ_i indicates an orientation of the eigenvector. So the combined effect of $I^{\pm}\Gamma$ is to judiciously orient and select the eigenspaces on which to pair the residues.

A traditional assignment problem can be used to calculate one or both of Γ and/or Ω. If we let $\xi_{i,j}$ be the "cost" of assigning entity i to entity j, the

classical assignment problem for a square ξ matrix is

$$\min \left\{ \sum_{i,j} \xi_{i,j}\omega_{i,j} : \sum_j \omega_{i,j} = 1 \; \forall i, \; \sum_i \omega_{i,j} = 1 \; \forall j, \; \omega_{i,j} \in \{0,1\} \right\}. \qquad (7)$$

To compute Γ we let $\xi_{i,j} = |\lambda'_i - \lambda''_j|$, which encourages eigenvectors with similar eigenvalues to be paired. Since the proteins are of different sizes, we replace $\sum_j \omega_{i,j} = 1$ with $\sum_j \omega_{i,j} \leq 1$. As with the square case, solving the problem is well known to be polynomial. We used the Hungarian algorithm in [5] to calculate Γ. To calculate Ω we let

$$\xi_{i,j} = \|[D'U^T]_{:,i} - [I^{\pm}\Gamma D''W^T]_{:,j}\|_p^p.$$

We further replace $\sum_i \omega_{i,j} = 1$ with $\sum_i \omega_{i,j} \leq 1$ and add $\sum_{i,j} \omega_{i,j} = S$, where S the maximum value in (5). This problem was modeled in AMPL and solved with CPLEX due to the changed constraints. Assignment problems were similarly used in [24].

The residue pairings from our initial numerical effort were disappointing in their biological measures. The problem was in the use of the assignment problem to calculate Ω, which was inadequate in its flexibility to handle gaps in the residue pairing. Gaps are controlled by S in the assignment problem. We used the equality $\sum_{i,j} \omega_{i,j} = S$, with S being the largest possible value, to guarantee a match between as many residues as possible. However, this assumption is not biologically sound. As an alternative, we compared the assignment method with a dynamic programming (DP) approach that pairs the residues. The DP algorithm is a standard global sequence alignment procedure [13] that allows, but penalizes, gaps in the alignment. This permits S to deviate from its maximum value. Secondary structure mismatches are also allowed but penalized. We refer interested readers to see [13] for a description of the procedure.

5 Numerical Results

We tested our algorithm's ability to identity the known families identified by SCOP [3] among 33 protein structures taken from the Skolnick data set [2, 6], see Table 2. The protein structures in the Skolnick data set were obtained from the Protein Data Bank [4] and parsed with BioPython [7]. The contact matrices were constructed with the piecewise linear sigmoid function mentioned in Section 2 with $\rho = \kappa = 7$. Other sigmoid functions were tested, but the piecewise linear function worked well with these parameters. Both the assignment method and the DP method were tested to calculate the permutation matrix Ω. The RMSD scores of our residue pairings were consistently worse for the assignment method, with an average improvement of 6.2% with DP in both the 1 and 2-norm objectives. For this reason the results below are based on the DP method for calculating Ω. Each gap in the residue alignment was penalized with a value of 2, and pairing residues from different secondary structures was

SCOP Fold	SCOP Family	Proteins
Flavodxin-like	CheY-related	1b00, 1dbw, 1nat, 1ntr, 3chy 1qmp(A,B,C,D), 4tmy(A,B)
Cupredoxin-like	Plastocyanin azurin-like	1baw, 1byo(A,B), 1kdi, 1nin 1pla, 2b3i, 2pcy, 2plt
TIM beta/alpha-barrel	Triosephosphate isomerase (TIM)	1amk, 1aw2, 1b9b, 1btm, 1hti 1tmh, 1tre, 1tri, 1ydv, 3ypi, 8tim
Ferritin-like	Ferritin	1b71, 1bcf, 1dps, 1fha, 1ier, 1rcd
Microbial ribonuclease	Fungal ribonucleases	1rn1(A,B,C)

Table 2. The Skolnick Data Set

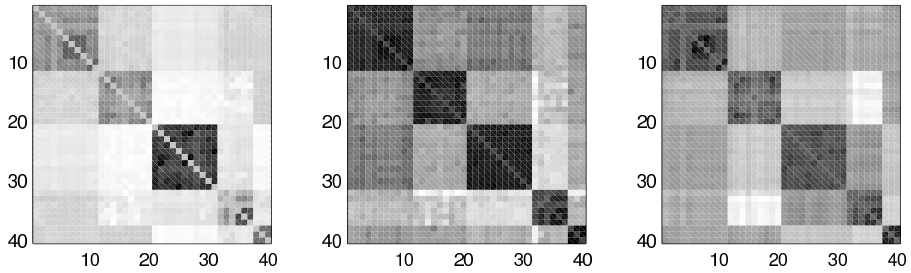
penalized with a value of 3.5. These parameters can be altered to remove/limit either gaps or mismatches. In our numerical work these values gave a mixture of gaps and mismatches.

Our algorithm was run on a dual core 2.16 GHz T2600 Intel processor with 1GB of memory in Matlab under Linux. The algorithm took 555.76 seconds to align 780 pairs of proteins with the 2-norm and 734.59 seconds with the 1-norm, approximately 0.71 seconds and 0.94 seconds per alignment, respectively. Andonov et al. [2] report a time of approximately 1.04 seconds per alignment for their algorithm on a 2.4 GHz AMD Opteron processor with 4 GB of memory programmed in C++. Computation of the eigensystem of each protein is not included as this is a one time operation. For small proteins the cost of computing the eigensystem of the protein's contact matrix is negligible, but the cost grows quickly for larger proteins. The eigensystems for large proteins should be computed once and stored for data base searches.

The graphs in Figure 3 depict the clustering ability of three different scores of our alignments with the 2-norm objective function. The first two scoring functions are widely used to assess protein alignments. STRUCTAL [23] has been reported to be a good scoring function for protein alignment [17] and is given by

$$\mathsf{STRUCTAL} = \sum_i \frac{20}{1 + \left(\frac{d_i}{2.24}\right)^2} - 10n_g.$$

The quantity d_i is the distance (after the structures have been superimposed) between the ith paired residues/atoms. The quantity n_g equals the total number of gaps in the alignment. The second scoring function is the RMSD of the aligned residues [14, 16], which is shown in Figure 3(b). Figure 3(c) is the score from the DP construction of Ω. Our algorithm correctly identifies the known 5 families in each case. RMSD most clearly distinguishes the families, with our DP values doing nearly as well. The STRUCTAL measure correctly identifies the families, although the delineations are not as sharp (especially for the 4th group).



(a) STRUCTAL (b) RMSD (c) DP Values

Fig. 3. Various scores for our alignments of the Skolnick data set with the 2-norm version of our objective function. The 40 proteins compared are ordered as they are listed in Table 2.

6 Conclusion

The eigensystem-based protein structure alignment algorithm described in this article is a new and fast way to align protein structures. The geometry of aligning the intrinsic contact vectors of two proteins provides additional insight into the protein alignment problem. This geometric interpretation of the problem is not available from the contact map overlap problem formulation which has a more graph-theoretic flavor. By solving an assignment problem, we can quickly pair the eigenvalues of the contact matrices of two proteins. Once an orientation for the second protein's eigenvectors has been specified, the corresponding eigenvectors are easily paired, providing a quick, permutation independent way to compare two protein structures. The key challenge solved in this paper is a method for quickly identifying a good orientation for the eigenvectors of the second protein. The last step in the alignment is to solve a standard global sequence alignment problem. Because this alignment is done only once, the algorithm is fast, at least comparable in speed to the latest algorithms for the contact map overlap problem, but with the potential to scale well for larger problems.

Acknowledgments

The idea of using the eigensystem of protein contact maps to align protein structures was inspired by the work of Galaktionov and Marshall [9] on protein structure prediction. Some preliminary ideas for the algorithm described in this paper were developed during the first author's 2005-06 sabbatical visit, hosted by Garland Marshall, at the Center for Molecular Design, Washington University, St. Louis, Missouri, USA.

References

1. Pankaj K Agarwal, Nabil H Mustafa, and Yusu Wang. Fast molecular shape matching using contact maps. *J Comput Biol*, 14(2):131–143, Mar 2007.
2. Rumen Andonov, Nicola Yanev, and Noël Malod-Dognin. An efficient lagrangian relaxation for the contact map overlap problem. In *WABI '08: Proceedings of the 8th international workshop on Algorithms in Bioinformatics*, pages 162–173, Berlin, Heidelberg, 2008. Springer-Verlag.
3. Antonina Andreeva, Dave Howorth, John-Marc Chandonia, Steven E Brenner, Tim J P Hubbard, Cyrus Chothia, and Alexey G Murzin. Data growth and its impact on the scop database: new developments. *Nucleic Acids Res*, 36(Database issue):D419–D425, Jan 2008.
4. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Res*, 28(1):235–242, Jan 2000.
5. Yi Cao. Hungarian algorithm for linear assignment problems (v2.1), 2008. http://www.mathworks.com/matlabcentral/fileexchange/20652.
6. Alberto Caprara, Robert Carr, Sorin Istrail, Giuseppe Lancia, and Brian Walenz. 1001 optimal pdb structure alignments: integer programming methods for finding the maximum contact map overlap. *J Comput Biol*, 11(1):27–52, 2004.
7. Peter J A Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J L de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, Jun 2009.
8. R. J. Forrester and H. J. Greenberg. Quadratic binary programming models in computational biology. *Algorithmic Operations Research*, 3:110–129, 2008.
9. S. G. Galaktionov and G. R. Marshall. Prediction of protein structure in terms of intraglobular contacts: 1d to 2d to 3d. Fourth International Conference on Computational Biology, Intelligent Systems for Molecular Biology '96,St. Louis, Missouri, U.S.A., June 12–15 1996.
10. A. Godzik. The structural alignment between two proteins: is there a unique answer? *Protein Sci*, 5(7):1325–1338, Jul 1996.
11. D. Goldman, S. Istrail, and C.H. Papadimitriou. Algorithmic aspects of protein structure similarity. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 512–521, 1999.
12. T. F. Havel, I. D. Kuntz, and G. M. Crippen. The combinatorial distance geometry method for the calculation of molecular conformation. i. a new approach to an old problem. *J Theor Biol*, 104(3):359–381, Oct 1983.
13. Neil C. Jones and Pavel A. Pevzner. *An Introduction to Bioinformatics Algorithms*. MIT Press, 2004.
14. W Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallog A*, 34:827–828, 1978.
15. Wolfgang Kabsch and Chris Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.
16. SK Kearsley. On the orthogonal transformation used for structural comparisonsh.m.berman, j.westbrook, z.feng, g.gilliland, t.n.bhat, h.weissig, i.n.shindyalov, p.e.bourne. *Acta Crystallog A*, 45:208–210, 1989.

17. Rachel Kolodny, Patrice Koehl, and Michael Levitt. Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J Mol Biol*, 346(4):1173–1188, Mar 2005.
18. Rachel Kolodny and Nathan Linial. Approximate protein structural alignment in polynomial time. *Proc Natl Acad Sci U S A*, 101(33):12201–12206, Aug 2004.
19. G. Lancia, R. Carr, and B. Walenz adn S. Istrail. 101 optimal pdb structure alignments: A branch-and-cut algorithm for the maximum contact map overlap problem. In *Proceedings of the Fifth Annual International Conference on Computational Biology*, pages 143–202, New York, NY, 2001. ACM Press.
20. Matthew Menke, Bonnie Berger, and Lenore Cowen. Matt: local flexibility aids protein multiple structure alignment. *PLoS Comput Biol*, 4(1):e10, Jan 2008.
21. Mark T Oakley, Daniel Barthel, Yuri Bykov, Jonathan M Garibaldi, Edmund K Burke, Natalio Krasnogor, and Jonathan D Hirst. Search strategies in structural bioinformatics. *Curr Protein Pept Sci*, 9(3):260–274, Jun 2008.
22. Dawn M. Strickland, Earl Barnes, and Joel S. Sokol. Optimal protein structure alignment using maximum cliques. *Oper. Res.*, 53(3):389–402, 2005.
23. S. Subbiah, D. V. Laurents, and M. Levitt. Structural similarity of dna-binding domains of bacteriophage repressors and the globin core. *Curr Biol*, 3(3):141–148, Mar 1993.
24. Y. Wang, F. Makedon, J. Ford, and H. Huang. A bipartite graph matching framework for finding correspondences between structural elements in two proteins. In *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, volume 2, pages 2972–2975, Sept. 2004.
25. Wei Xie and Nikolaos V Sahinidis. A reduction-based exact algorithm for the contact map overlap problem. *J Comput Biol*, 14(5):637–654, Jun 2007.