8-6-2010

A Spectral Approach to 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

MSTR 10-06

A Spectral Approach to Protein Structure Alignment

Yosi Shibberu, Allen Holder

Mathematical Sciences Technical Report Series
MSTR 10-06

August 6, 2010

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

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

A Spectral Approach to Protein Structure Alignment

Yosi Shibberu, Allen Holder

Mathematics Department

Rose-Hulman Institute of Technology

Terre Haute, IN 47803

(Invited Paper)

Index Terms

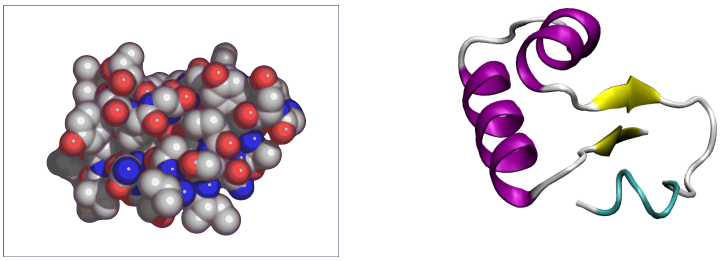
protein structure alignment, structural bioinformatics, contact maps, spectral methods

Abstract

We present two algorithms that use spectral methods to align protein folds. One of the algorithms is suitable for database searches, the other for difficult alignments. We present computational results for 780 pairwise alignments used to classify 40 proteins as well as results for a separate set of 36 protein alignments used for comparison to four other alignment algorithms. We also provide a mathematically rigorous development of the intrinsic geometry underlying our spectral approach.

1 Introduction

Proteins are long molecular chains constructed from twenty amino acids (residues) and are an important part of most biochemical processes. Protein chains fold into unique, tightly packed globular structures called folds. See Figure 1. The particular sequence of amino acids determines the proteins unique fold, and the geometry of a proteins fold largely determines its specific biological function. Identifying the function of an individual protein is an important and challenging problem. A better understanding



(a) protein atoms (b) protein chain

Fig. 1: (a) The 3D geometry of Crambin (1CRN), a small protein consisting of 46 residues and 327 atoms (not including hydrogen atoms). (b) The 3D geometry of a fold is often represented by a cartoon depicting the path of the protein's chain through its fold.

of protein evolution would aid the identification of protein function and could lead to advances in biology as well as new treatments for diseases.

The evolution of proteins is studied by making comparisons, either by aligning protein sequences or aligning protein folds. Fold-based comparisons are believed to be more informative and robust [23]. The question of how to achieve fast, accurate fold-based alignments continues to be a topic of current interest [13], [27], [28], [24], [32], [1], [20].

The objective in protein alignment is to determine a one-to-one correspondence between a subset of the residues in two different protein folds. (See Figure 2 for a two dimensional version of the problem.) The subset chosen should optimize some biologically relevant similarity measure, although there is currently no consensus on what this measure of similarity should be [23], [13].

In protein alignment, the ordering of the residues in a protein's chain plays a key role in making comparisons. This is largely for biological reasons. Nature assembles proteins from DNA in a linear fashion. Over the course of time, mutations occur in DNA causing corresponding changes in a protein's chain. The changes are either insertions, deletions or substitutions of individual residues of the protein's chain. It is reasonable to assume that as proteins evolve over time, they preserve the sequence order of the segments of their chains [7]. Thus, we only consider alignments that preserve the sequence order of each protein's chain.

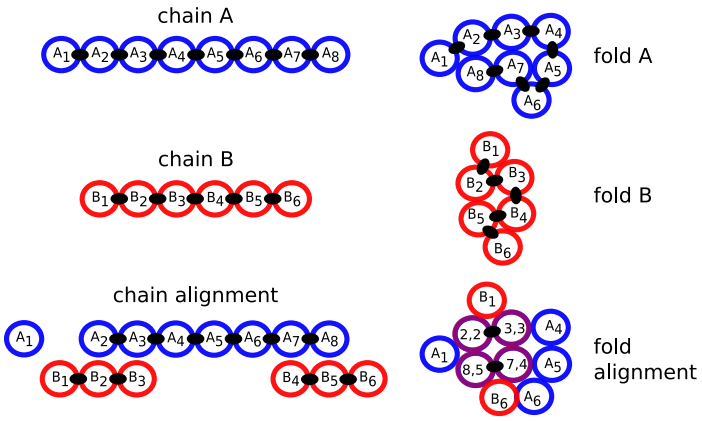
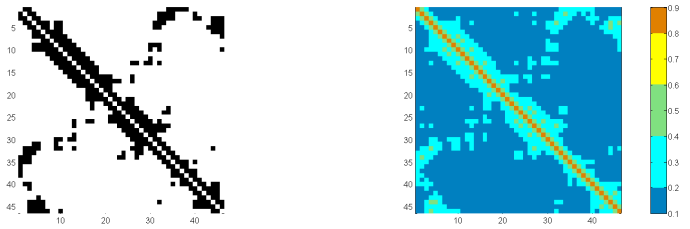


Fig. 2: A two dimensional version of the protein alignment problem. Aligning protein A to protein B involves gapping the protein chains and pairing the remaining atoms so that the chain order is preserved in each. The alignment shown is $(A_2, B_2), (A_3, B_3), (A_7, B_4), (A_8, B_5)$. The indices of each of the atoms in the alignment must increase. Note, (A_2, B_2) is indicated in the fold alignment by $2, 2$. Other aligned pairs are denoted similarly.

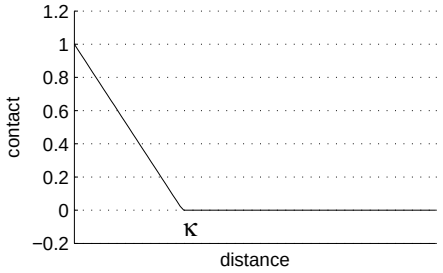
A protein's fold is normally described by the three dimensional (3D) Cartesian coordinates of the protein's atoms. A distance matrix specifying all the distances between pairs of atoms completely determines the fold up to reflections in a coordinate invariant way [14], [9]. A distance matrix is often converted into a contact matrix whose entries are equal to one for residues that are within a certain cutoff distance from one another and zero otherwise. See Figure 3(a). We use a piecewise-linear, continuous cutoff function (Figure 4) to obtain a smoothed contact matrix like the one in Figure 3(b). In particular, we can choose our cutoff function so that the smoothed contact matrix we obtain is positive definite. We then use this matrix to define an N dimensional Euclidean space to represent the fold, where N equals the number of residues in the protein's chain. In this representation of a fold, residues are associated with unit vectors we refer to as the intrinsic contact vectors of the fold. See Figure 5.

The underlying optimization problem associated with our spectral method was first



(a) sharp contact matrix (b) smoothed contact matrix

Fig. 3: Contact matrices for Crambin, the protein depicted in Figure 1.



$$\sigma(d_{ij}) = \begin{cases} 1 - \frac{1}{\kappa} d_{ij} & 0 \leq d_{ij} \leq \kappa \\ 0 & \text{otherwise} \end{cases}$$

Fig. 4: Piecewise-linear, continuous cutoff function with cutoff parameter κ.

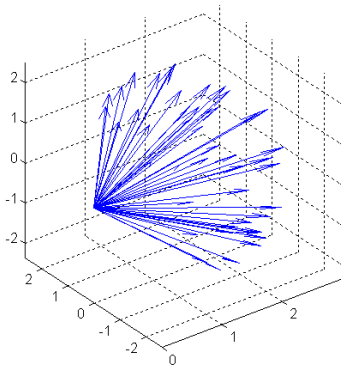


Fig. 5: The 46 intrinsic contact vectors of the fold of Crambin, the protein depicted in Figure 1. The vectors have been projected from 46 dimensional space to 3D space.

proposed in [26]. That paper develops a heuristic to orient the spectral information in a way that bounds the maximum deviation between the two proteins with respect to their smooth contact maps. The heuristic was shown to be efficient for classification purposes. In this work, we give a rigorous development of the intrinsic geometry associated with the optimization problem and develop several new heuristics based on spectral information. These heuristics are compared with the earlier heuristic in [26] and with other published methods. We show that our general algorithm achieves monotonic convergence in contact space. Two data sets are used to compare our methods on problems of varying difficulty. Our numerical results are promising, and in particular, our fastest alignment method appears suitable for database-wide alignments.

2 CONTACT GEOMETRY

Let the columns of X equal $(x_i, y_i, z_i)^\top,\ i = 1, 2, \ldots, N$, so that column i contains the 3D coordinates of the ith C_α atom of a protein. (The C_α atoms are numbered consecutively along a protein chain.) Define the C_α distance matrix of a protein to be the matrix $\mathrm{D} = [d_{ij}]$ where d_{ij} is the Euclidean distance between the ith and jth $C\alpha$ atoms. We represent the geometry of a fold by a C_α contact matrix $\mathrm{C} = [c_{ij}]$, where C is computed from the distance matrix D by applying a cutoff function, $c_{ij} = \sigma(d_{ij})$, like the one given in equation (1). We define the contact between residue i and residue j of a protein to be c_{ij}.

$$\sigma(d_{ij}) = \begin{cases} 1 - \frac{1}{\kappa} d_{ij} & 0 \le d_{ij} \le \kappa \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

If the cutoff parameter, κ, in equation (1), is chosen sufficiently small, the contact matrix C is diagonally dominant and hence positive-definite [26]. The positive-definiteness of C plays a key role in our geometric formulation of the protein alignment problem because it allows us to represent a protein fold in N-dimensional Euclidean space, where N is the length of the chain, i.e. the number of residues. Specifically, the contact matrix C defines the generalized inner product

$$\langle \mathrm{u}, \mathrm{v} \rangle_{\mathrm{C}} = \mathrm{u}^\top \mathrm{C} \mathrm{v}. \tag{2}$$

This inner product has a useful interpretation. If we assign residue i to the standard unit vector $\mathrm{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$, then the contact between residue i and residue j is given by the inner product

$$\langle \mathrm{e}_i, \mathrm{e}_j \rangle_{\mathrm{C}} = \mathrm{e}_i^\top \mathrm{C} \mathrm{e}_j = c_{ij}.$$

In support of this observation, we define e_i to be the *standard contact coordinates* of residue i. (We refer to the C_α atom coordinates $(x_i, y_i, z_i)^\top$ as the 3D coordinates of residue i.) The discussion above motivates the following definition.

Definition 1 (Contact Space): A contact space is an N-dimensional Euclidean space $\mathbb{R}^N$ with generalized inner product $\langle \cdot, \cdot \rangle_{\mathrm{C}}$, where C is a positive-definite contact matrix.

Before we align the folds of two proteins, we first define an appropriate coordinate system in which to make comparisons. The order of the standard contact coordinate system $\{\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_N\}$ is determined by the sequence of residues in a protein's chain. The path of a protein chain through a given fold, however, varies from protein to protein. It makes sense to define a new coordinate system that is independent of the order of the residues in a protein's chain and that is intrinsic to the fold itself. We do this by solving a sequence of optimization problems as shown below.

Let the first unit vector in our intrinsic coordinate system be the unit vector v_1 that has the largest sum of squared contacts with all the residues in a fold. The contacts v_1 has with each residue is given by the vector

$$\mathrm{w} = \langle (\mathrm{e}_1, \ldots, \mathrm{e}_n), \mathrm{v}_1 \rangle_{\mathrm{C}} = \mathrm{I}^\top \mathrm{C} \mathrm{v}_1 = \mathrm{C} \mathrm{v}_1, \tag{3}$$

and the sum of the squared contact is $s = \mathrm{w}^\top \mathrm{w} = \mathrm{v}_1^\top \mathrm{C}^\top \mathrm{C} \mathrm{v}_1$. Therefore, v_1 is the solution to the following optimization problem:

$$\begin{aligned} \max \quad & \mathrm{v}_1^\top \mathrm{C}^\top \mathrm{C} \mathrm{v}_1 \\ \text{subject to} \quad & \mathrm{v}_1^\top \mathrm{v}_1 = 1. \end{aligned} \tag{4}$$

The second unit vector is defined similarly, as it is the unit vector, v_2, that has the largest sum of squared contacts, but with the additional constraint that v_2 must be perpendicular to v_1. Therefore, it is a solution to

$$\begin{aligned} \max \quad & \mathrm{v}_2^\top \mathrm{C}^\top \mathrm{C} \mathrm{v}_2 \\ \text{subject to} \quad & \mathrm{v}_2^\top \mathrm{v}_2 = 1, \ \mathrm{v}_2^\top \mathrm{v}_1 = 0. \end{aligned} \tag{5}$$

Proceeding in this fashion we construct a contact coordinate system, $\{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_N\}$, that is intrinsic to a particular fold and that is independent of the order of the residues in a given protein's chain. A standard result from linear algebra known as Rayleigh's Principle [22] implies that $\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_N$ are equal to the eigenvectors of the contact matrix C.

Theorem 1 (Rayleigh's Principle): Assume C is an N by N, symmetric, positive definite matrix. The solution to the sequence of optimization problems

$$\begin{aligned} \max \quad & \mathrm{v}_i^\top \mathrm{C}^\top \mathrm{C} \mathrm{v}_i \\ \text{subject to} \quad & \mathrm{v}_1^\top \mathrm{v}_i = 0, \ldots, \mathrm{v}_{i-1}^\top \mathrm{v}_i = 0, \mathrm{v}_i^\top \mathrm{v}_i = 1 \end{aligned}$$

for $i = 1, \ldots, N$ is given by the eigenvectors of C.

The following lemma shows that the eigenvalue, λ_i, associated with eigenvector v_i, is a measure of the contact between v_i and the entire fold.

Lemma 1: The square-root of the sum of all the squared contacts that the unit vectors, e_j, $j = 1, \ldots, N$ make with eigenvector v_i equals the eigenvalue λ_i associated with v_i.
Proof: Let $\mathrm{w} = \langle (\mathrm{e}_1, \ldots, \mathrm{e}_N), \mathrm{v}_j \rangle_{\mathrm{C}} = \mathrm{C}\mathrm{v}_i$. Then the square root of the squared contacts that the standard unit vectors make with eigenvector v_i equals

$$\sqrt{\mathrm{w}^\top \mathrm{w}} = \sqrt{\mathrm{v}_i^\top \mathrm{C}^\top \mathrm{C} \mathrm{v}_i} = \sqrt{\lambda_i^2} = \lambda_i.$$

□

Taken together, Theorem 1 and Lemma 1 imply that the intrinsic contact coordinate system $\{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_N\}$ is ordered (from largest to smallest) by the size of the eigenvalues, $\lambda_1, \lambda_2, \ldots, \lambda_N$ of the contact matrix C. If the eigenvalues of a fold are distinct (which has always been observed to be the case) the corresponding eigenspaces are one dimensional and the eigenvectors defined by Rayleigh's Principle are unique up to orientation. This means the ordering is unique for practical purposes. Moreover, this order is independent of the sequence of a particular chain through a fold and is intrinsic to the fold itself. We refer to the ordered set of eigenvalues of a contact matrix of a fold as the *spectrum* of the fold.

Given a vector, x, in standard coordinates, we compute its intrinsic coordinates, y, by solving the equation $\mathrm{x} = \mathrm{V}\mathrm{y}$, where $\mathrm{V} = [\mathrm{v}_1 | \mathrm{v}_2 | \ldots | \mathrm{v}_N]$ and where $\mathrm{v}_i,\ i = 1, 2, \ldots, N$ are

intrinsic unit vectors given by Rayleigh's Principle. Since V is an orthonormal matrix, $\mathrm{y} = \mathrm{V}^\top \mathrm{x}$. Under the mapping $\mathrm{V}^\top : \mathrm{x} \mapsto \mathrm{y}$, the inner product $\langle \cdot, \cdot \rangle_\mathrm{C}$ maps to $\langle \cdot, \cdot \rangle_\mathrm{D}$, where D is a diagonal matrix with the eigenvalues of C along the main diagonal. Suppose we are interested in comparing protein A and B with contact maps C_A and C_B respectively. In intrinsic coordinate, we will, in general, have two distinct inner products, $\langle \cdot, \cdot \rangle_{\mathrm{D}_A}$ and $\langle \cdot, \cdot \rangle_{\mathrm{D}_B}$. We must therefore align the spectrum of the folds before we define a common contact space to make comparisons. (Spectrum alignment is the bases for the fast fold alignment algorithm described in section 3.)

Before mapping a vector to intrinsic contact space, it is convenient to apply the scale transformation $\sqrt{\mathrm{D}} : \mathrm{x} \mapsto \mathrm{y}$, which is allowed since the contact maps are assumed to be positive definite. This allows us to use the standard inner product $\langle \cdot, \cdot \rangle$ in intrinsic contact space, as we prove in Theorem 2. First, we define intrinsic contact coordinates.

Definition 2 (Intrinsic Contact Coordinates): Let $\mathrm{C} = \mathrm{VDV}^\top$ be the eigenvalue-eigenvector decomposition of the contact matrix C with eigenvalues sorted from largest to smallest in size. Define $\mathrm{R} = \sqrt{D}\mathrm{V}^\top$. The columns of R are the intrinsic contact coordinates of a fold with contact matrix C.

Since $\mathrm{r}_i = \mathrm{Re}_i$ is a column of R, the matrix R maps the standard contact coordinates, e_i, of residue i, to its intrinsic contact coordinates, r_i. The next theorem shows that R maps the generalized inner product $\langle \cdot, \cdot \rangle_\mathrm{C}$ to the standard inner product $\langle \cdot, \cdot \rangle$.

Theorem 2: The matrix $\mathrm{R} = \sqrt{D}\mathrm{V}^\top$ maps the generalized inner product $\langle \cdot, \cdot \rangle_\mathrm{C}$ to the standard inner product $\langle \cdot, \cdot \rangle$.

Proof: It is sufficient to prove that the result holds true for the standard contact vectors $\{\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_N\}$ since they form a basis for contact space. For $i, j = 1, 2, \ldots, N$, we have

$$\begin{aligned}
\langle \mathrm{e}_i, \mathrm{e}_j \rangle_\mathrm{C} &= \mathrm{e}_i^\top \mathrm{C} \mathrm{e}_j \\
&= \mathrm{e}_i^\top \mathrm{VDV}^\top \mathrm{e}_j \\
&= \mathrm{e}_i^\top \left(\sqrt{\mathrm{D}} \mathrm{V}^\top \right)^\top \left(\sqrt{\mathrm{D}} \mathrm{V}^\top \right) \mathrm{e}_j \\
&= \left(\mathrm{Re}_i \right)^\top \left(\mathrm{Re}_j \right) \\
&= \mathrm{r}_i^\top \mathrm{r}_j \\
&= \langle \mathrm{r}_i, \mathrm{r}_j \rangle.
\end{aligned}$$

□

Note that even with the scale transformation, the intrinsic contact vectors r_j, $j = 1, 2, \ldots, N$, are unit vectors since $\langle \mathrm{r}_j, \mathrm{r}_j \rangle = \langle \mathrm{e}_j, \mathrm{e}_j \rangle_{\mathrm{C}} = c_{jj} = 1$. Note also that R preserves the orthogonality of the eigenvectors v_i but not their lengths, since $\mathrm{R}\mathrm{v}_i = \sqrt{\mathrm{D}}\mathrm{V}^\top \mathrm{v}_i = \sqrt{\lambda}\mathrm{e}_i$. The *contact length* of v_i is given by

$$\|\mathrm{v}_i\|_{\mathrm{C}} = \sqrt{\langle \mathrm{v}_i, \mathrm{v}_i \rangle_{\mathrm{C}}} = \sqrt{\langle \sqrt{\lambda_i}\mathrm{e}_i, \sqrt{\lambda_i}\mathrm{e}_i \rangle} = \|\sqrt{\lambda_i}\mathrm{e}_i\| = \sqrt{\lambda_i}, \tag{6}$$

which is preserved by R.

The intrinsic contact geometry of a fold has a simple geometric interpretation. The contact between any two residues, i and j, is $\langle \mathrm{r}_i, \mathrm{r}_j \rangle = \cos(\theta)$, where θ is the angle between their intrinsic contact vectors. Residues that are not in contact have orthogonal contact vectors. Since all contacts are greater than or equal to zero, all the contact vectors of a fold are within 90° of each other. We caution, however, that when we align two different folds, we may have protein-protein contacts that are negative.

3 Spectrum Alignment

The spectrum $\lambda_1, \lambda_2, \ldots, \lambda_N$ of a fold is somewhat like a finger print and can be use to classify folds. The spectrum, however, does not contain information on the path of a protein chain through its fold. We present an algorithm that takes this information into account and constructs a crude, but fast, fold alignment.

Our algorithm has a preprocessing step that partitions the residues of a fold by eigenspace. Specifically, we assign each residue to the eigenspace of the fold the residue is closest to in terms of *contact angle*. Theorem 3 provides a formula for computing the cosine of this contact angle. Each residue is then assigned the eigenvalue of its eigenspace. Finally, we compute the matrix $\mathrm{S} = [s_{ij}]$ with $s_{ij} = |\lambda_i - \lambda_j|$ to score alignments and apply dynamic programming to compute the optimal alignment.

Dynamic programming (DP), first used by Needleman and Wunsch to align proteins [21], is commonly used in bioinformatics to align sequences. Apart from being relatively fast and easy to implement, DP based sequence alignments have the biologically desirable property of preserving the sequence-order of the sequences aligned. DP sequence

alignment require as input: 1) an alignment scoring matrix and 2) a gap penalty. The scoring matrix gives the score of aligning element A_i of the first sequence with element B_j of the second sequence. The gap penalty penalizes the creation of gaps in the alignment. (The interested reader is referred to [16], [5], [10] for a detailed description of DP and sequence alignment.)

In database applications, a protein may be aligned to thousands of other proteins. The preprocessing step of our algorithm is completed only once per protein and the eigenvalues of the residues in each fold are then stored in the database. During a search, only the scoring matrix, S, needs to be computed and dynamic programming applied using this matrix to compute an alignment.

We next provide a detailed description of our algorithm outline in Table 1. First we show how to compute the cosine of the contact angle between the contact vector of a residue and an eigenspace of a fold.

Theorem 3 (Residue-Eigenspace Contact Angle): The cosine of the contact angle, θ_{ij}, between residue j and eigenspace i, equals $\sqrt{\lambda_i}v_{ji}$ where v_{ji} is the jth component of eigenvector v_i.

Proof: The proof is more intuitive in standard coordinates. In standard coordinates we have that

$$\langle \mathrm{e}_j, \mathrm{v}_i \rangle_C = \mathrm{e}_j^\top \mathrm{C} \mathrm{v}_i = \lambda_i \mathrm{e}_j^\top \mathrm{v}_i = \lambda_i v_{ji}. \tag{7}$$

But,

$$\langle \mathrm{e}_j, \mathrm{v}_i \rangle_C = \|\mathrm{e}_j\|_C \|\mathrm{v}_i\|_C \cos(\theta_{ij}) = \sqrt{\lambda_i} \cos(\theta_{ij}). \tag{8}$$

Therefore, $\lambda_i v_{ji} = \sqrt{\lambda_i} \cos(\theta_{ij})$, implying that $\cos(\theta_{ij}) = \sqrt{\lambda_i} v_{ji}$.

□

Recall that column j of $R = \sqrt{\mathrm{D}}\mathrm{V}^\top$ is the intrinsic contact coordinates of residue j. Coordinate i of residue j is equal to $\sqrt{\lambda_i}v_{ji}$, which, by Theorem 3 equals $\cos(\theta_{ij})$, where θ_{ij} is the angle between residue j and eigenspace i. Therefore, we assign residue j, eigenvalue λ_{i^*}, were i^* is the solution to $\max_i \sqrt{\lambda_i}|v_{ji}|$. This is accomplished by determining the row, i^*, in which the maximum of column j of R occurs.

1) Preprocessing:
   a) Construct the contact matrix C of each fold.
   b) Compute the eigensystem decomposition $\mathrm{C} = \mathrm{VDV}^\top$ of each fold.
   c) Compute the intrinsic contact coordinates $\mathrm{R} = \sqrt{D}\mathrm{V}^\top$ of each fold.
   d) Determine the mapping $\nu : j \mapsto i^*$, which gives the row in which the maximum of each column of $|R|$ occurs.
   e) Assign residue j eigenvalue $\lambda_{\nu(j)}$.
2) Construct alignment scoring matrix $S = [s_{ij}]$ where $s_{ij} = |\lambda_i - \lambda_j|$.
3) Align folds using DP with scoring matrix S and a suitable gap penalty.

TABLE 1: Spectrum Alignment Algorithm

4 Protein Classification

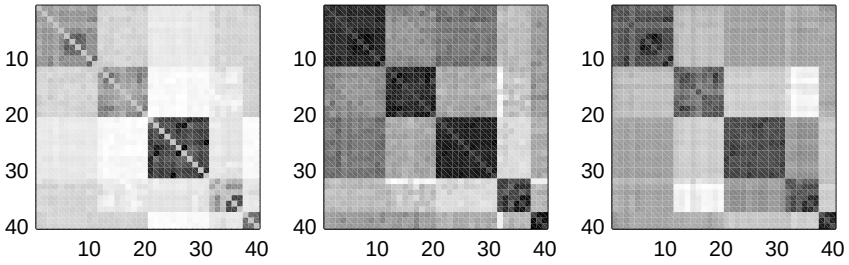
To test our spectrum algorithm, we repeated the protein classification computations reported in our earlier work [26]. We evaluated the ability of the spectrum algorithm to identify the known families identified by SCOP [2] among 40 protein structures taken from the Skolnick data set [1], [6] given in Table 2. The protein structures in the Skolnick data set were obtained from the Protein Data Bank [3] and parsed with BioPython [8]. A cutoff value of $\kappa = 8$ Angstroms and a gap penalty equal to 1 was used for the spectrum alignment method. The delta alignment method tested in our earlier work required 553.3 seconds to compute 780 alignments or approximately 0.7 seconds per alignment. The spectrum alignment algorithm required only 96.3 seconds, 0.12 seconds per alignment, an 83% reduction in processing time. The preprocessing times for both methods are not included as this step is done only once when a database is created. Figure 6 indicates that the spectrum algorithm does a good job of correctly clustering the protein families in the Skolnick data set.

5 Contact Alignment

A difficult step in our approach to aligning proteins is constructing a good alignment scoring matrix. We use protein-protein contact matrices in this step of the alignment

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 1thm, 1tre, 1tri, 1ydv, 3ypi, 8tim
Ferritin-like	Ferritin	1b71, 1bcf, 1dps, 1fha, 1ier, 1rcd
Microbial ribonuclease	Fungal ribonucleases	1rn1(A,B,C)

TABLE 2: Skolnick data set.



(a) STRUCTAL (b) RMSD (c) Contact Score

Fig. 6: Alignment scores for the Skolnick data set. The 40 proteins compared are ordered as they are listed in Table 2. (a) STRUCTAL is a widely used alignment scoring function [30], [18]. (b) RMSD is computed after the structures have been superimposed using the Kabsch procedure. (c) Contact Score is the score maximized by DP.

process. The eigenspaces of the two proteins must be aligned and oriented before a protein-protein contact matrix can be computed. (Note that aligning eigenvectors is simpler than aligning residues because unlike intrinsic contact vectors, eigenvectors are orthogonal.) We then compute a protein-protein contact matrix by computing all of the inner products of the intrinsic contact vectors of the first protein with the second protein. Finally, we use DP to compute an alignment.

Specifically, let C_A and C_B be the contact matrices of proteins A and B. Using the

procedure outlined in section 2, we compute intrinsic contact coordinates, R_A and R_B for each protein fold. Since the protein chains typically have different lengths, we determine two orthogonal projection matrices, Γ_A and Γ_B, that gap the spectrum of each protein's fold. We also determine a permutation matrix Ω that appropriately pairs the gapped spectrum of each fold. Finally, we determine a diagonal matrix $\mathrm{I}^{\pm}$ with ± 1 along the diagonal, that appropriately orients the eigenspaces of one of the folds. Let $\hat{\mathrm{R}}_A$ and $\hat{\mathrm{R}}_B$ be the intrinsic contact coordinates after the eigenspaces of each fold have been appropriately gapped, paired and oriented, i.e., let $\hat{\mathrm{R}}_A = \Omega\Gamma_A\mathrm{R}_A$ and $\hat{\mathrm{R}}_B = \mathrm{I}^{\pm}\Gamma_B\mathrm{R}_B$. The protein-protein contact matrix is then $\mathrm{C}_{AB} = \hat{\mathrm{R}}_A^{\top}\hat{\mathrm{R}}_B$.

Since the spectrum of a fold has a natural ordering, we can use DP to determine $\Gamma_A,\ \Gamma_B$ and Ω which align the eigenspaces of proteins A and B. DP has the added advantage of preserving the sequence order of the spectrum alignment. We need an eigenspace-alignment scoring matrix, E_{AB} for this purpose. The residues of proteins A and B must be aligned before E_{AB} can be computed. This presents us with a "chicken-and-egg" problem which we solve by using the iteration algorithm outlined in Table 3. Before describing this algorithm in detail, we explain how we compute the eigenvector alignment matrix E_{AB}.

The matrix E_{AB} is a protein-protein contact matrix for eigenvalue-weighted eigenvectors. (We use eigenvalue-weight eigenvectors because we want an alignment that aligns the eigenvalues as well as the eigenspaces.) We use the standard inner product in standard coordinates to compute all inner products of eigenvalue-weighted eigenvectors of protein A with eigenvalue-weighed eigenvectors of protein B. To do this, we need matrices $\Gamma_A^s,\ \Gamma_B^s$ and Ω^s that align the residues, and hence, the standard contact coordinates of the two proteins. (Superscript s denotes the matrices are acting on the protein chain sequence of residues.)

The rows of R are eigenvalue-weighted eigenvectors expressed in standard coordinates. Recall that row i of R equals $\sqrt{\lambda_i}\mathrm{v}_i$ and that the contact length of row i is $\|\sqrt{\lambda_i}\mathrm{v}_i\|_{\mathrm{C}} = \lambda_i$, hence the name eigenvalue-weighted eigenvector. It follows then that $\mathrm{E}_{AB} = \tilde{\mathrm{R}}_A\tilde{\mathrm{R}}_B^{\top}$, where $\tilde{\mathrm{R}}_A = \mathrm{R}_A\Gamma_A^s\Omega^s$ and $\tilde{\mathrm{R}}_B = \mathrm{R}_B\Gamma_B^s$. Since the eigenspaces are not oriented, we use $|\mathrm{E}_{AB}|$ to score eigenspace alignments and then choose the orientation matrix $\mathrm{I}^{\pm}$ that maximizes the final alignment.

1) Use the spectrum alignment algorithm (Table 1) to determine initial chain-sequence gapping matrices $\Gamma_A^s,\ \Gamma_B^s$ and a chain-order preserving permutatation Ω^s that align the residues of the protein chains.
2) Compute $\tilde{\mathrm{R}}_A = \mathrm{R}_A\Gamma_A^s\Omega_s$ and $\tilde{\mathrm{R}}_B = \mathrm{R}_B\Gamma_B^s$.
3) Compute the protein-protein, eigenvalue-weighted, eigenvector contact matrix $\mathrm{E}_{AB} = \tilde{\mathrm{R}}_A\tilde{\mathrm{R}}_B^\top$.
4) Determine eigenspace gapping matrices $\Gamma_A,\ \Gamma_B$ and spectrum-order preserving permutation Ω that align the eigenspaces of the protein folds by solving the following optimization problem with DP:

$$\max_{\Gamma_A,\Gamma_B,\Omega} \mathrm{trace}(\Omega\Gamma_A|\mathrm{E}_{AB}|\Gamma_B).$$

5) Determine the orientation matrix $\mathrm{I}^\pm$ which satisfies the equation

$$\mathrm{diag}(\mathrm{I}^\pm) = \mathrm{sign}(\mathrm{diag}(\Omega\Gamma_A\mathrm{E}_{AB}\Gamma_B)).$$

6) Compute $\hat{\mathrm{R}}_A = \Gamma_A\mathrm{R}_A$ and $\hat{\mathrm{R}}_B = \mathrm{I}^\pm\Omega\Gamma_B\mathrm{R}_B$.
7) Compute the protein-protein contact matrix $\mathrm{C}_{AB} = \hat{\mathrm{R}}_A^\top\hat{\mathrm{R}}_B$.
8) Determine protein chain gapping matrices $\Gamma_A^s,\ \Gamma_B^s$ and chain-order preserving permutation Ω^s that align the protein chains by solving the following optimization problem with DP:

$$\max_{\Gamma_A^s,\Gamma_B^s,\Omega^s} \mathrm{trace}(\Omega^s\Gamma_A^s\mathrm{C}_{AB}\Gamma_B^s).$$

9) Return to step 2 and repeat until the algorithm converges.

TABLE 3: Contact Alignment Algorithm

Our contact alignment algorithm in Table 3 alternates between aligning the eigenspaces of protein folds using standard contact coordinates (steps 2–5) and aligning protein chains using intrinsic contact coordinates (steps 6–8). Theorem 4 establishes that the algorithm converges monotonically.

Theorem 4: The contact alignment algorithm converges monotonically.

Proof: First we establish upper bounds for the optimization problems in steps 4 and 8 in Table 3. We then show that the solution to each optimization problem must increase

monotonically, thus establishing convergence of the algorithm.

Using the notation $(\alpha_1, \beta_1), \ldots, (\alpha_{m_1}, \beta_{m_2})$ to denote the eigenspace alignment determined by $\Gamma_A^s,\ \Gamma_B^s$ and Ω^s, we have that

$$\begin{aligned} \text{trace}(\Omega\Gamma_A \mathrm{E}_{AB}\Gamma_B) &= \sum_{i=1}^{m_1} \mathrm{E}^{AB}_{\alpha_i\beta_i} && (9)\\ &= \sum_{i=1}^{m_1} \sqrt{\lambda^A_{\alpha_i}\lambda^B_{\beta_i}}(\mathrm{v}^A_{\alpha_i})^\top \mathrm{v}^B_{\beta_i}, && (10) \end{aligned}$$

where $0 \leq m_1 \leq \min(N_1, N_2)$. Taking absolute values we have

$$\begin{aligned} \left|\text{trace}(\Omega\Gamma_A \mathrm{E}_{AB}\Gamma_B)\right| &\leq \sum_{i=1}^{m_1} \sqrt{\lambda^A_{\alpha_i}\lambda^B_{\beta_i}}\left|(\mathrm{v}^A_{\alpha_i})^\top \mathrm{v}^B_{\beta_i}\right| && (11)\\ &\leq \sum_{i=1}^{m_1} \sqrt{\lambda^A_{\alpha_i}\lambda^B_{\beta_i}} && (12)\\ &\leq \sum_{i=1}^{\min(N_1,N_2)} \sqrt{\lambda^A_i\lambda^B_i}, && (13) \end{aligned}$$

where we have used the fact that the eigenvectors $\mathrm{v}^A_{\alpha_i}$ and $\mathrm{v}^B_{\beta_i}$ are unit vectors and that the eigenvalues λ^A_i and λ^B_i are ordered from largest to smallest. Inequality (13) establishes an upper bound for the optimization problem in step 4 of the algorithm.

Using the notation $(a_1, b_1), \ldots, (a_{m_1}, b_{m_1})$ to denote the protein chain alignment determined by the matrices $\Gamma_A^s,\ \Gamma_B^s$ and Ω^s we have that

$$\begin{aligned} \left|\text{trace}(\Omega^s\Gamma^s_A \mathrm{C}_{AB}\Gamma^s_B)\right| &\leq \sum_{j=1}^{m_2} \left|C^{AB}_{a_jb_j}\right| && (14)\\ &= \sum_{j=1}^{m_2} \left|(\mathrm{r}^A_{a_j})^\top \mathrm{r}^B_{b_j}\right| && (15)\\ &\leq \min(N_1, N_2), && (16) \end{aligned}$$

where we have used the fact that the intrinsic contact vectors r^A_j and r^B_j are unit vectors and the fact that $0 \leq m_2 \leq \min(N_1, N_2)$. Inequality (16) establishes an upper bound for the optimization problem in step 8 of the algorithm.

Next, we show that the solution to the optimization problems in steps 4 and 8 increase monotonically. We use the fact that for a given protein chain alignment $(a_1, b_1) \ldots (a_{m_2}, b_{m_2})$ and a given eigenvector alignment $(\alpha_1, \beta_1) \ldots (\alpha_{m_1}, \beta_{m_1})$ and orientation $\mathrm{I}^{\pm}$, the objective

functions of both optimization problems are equal. Specifically, we show that

$$\sum_{i=1}^{m_1} \mathrm{E}^{AB}_{\alpha_i\beta_i} = \sum_{j=1}^{m_2} \mathrm{C}^{AB}_{a_jb_j}. \tag{17}$$

However, for the moment, assume (17) is valid. Let $(\alpha_1^*, \beta_1^*) \ldots (\alpha_{m_1^*}^*, \beta_{m_1^*}^*)$ denote the globally optimal solution to the optimization problem in step 4 and let $(a_1^*, b_1^*) \ldots (a_{m_2^*}^*, b_{m_2^*}^*)$ denote the globally optimal solution to the optimization problem in step 8. Using equation (17), we have that

$$\begin{aligned}
\sum_{i=1}^{m_1} \mathrm{E}^{AB}_{\alpha_i\beta_i} &= \sum_{j=1}^{m_2} \mathrm{C}^{AB}_{a_jb_j} && (18)\\
&\leq \sum_{j=1}^{m_2^*} \mathrm{C}^{AB}_{a_j^*b_j^*} && (19)\\
&= \sum_{i=1}^{m_1} \left(\mathrm{E}^{AB}_{\alpha_i\beta_i}\right)^* && (20)\\
&\leq \sum_{i=1}^{m_1^*} \left(\mathrm{E}^{AB}_{\alpha_i^*\beta_i^*}\right)^*, && (21)
\end{aligned}$$

where the notation $\left(\mathrm{E}^{AB}_{\alpha_i\beta_i}\right)^*$ indicates that the eigenvector contact matrix is computed using the optimal protein chain alignment $(a_1^*, b_1^*) \ldots (a_{m_2^*}^*, b_{m_2^*}^*)$. Inequality (21) shows that the solution to the optimization problem in step 4 increases monotonically. A similar argument establishes that the solution to the optimization problem in step 8 also increases monotonically.

To complete the proof, we establish equation (17), which follows since

$$\begin{aligned}
\sum_{i=1}^{m_1} \mathrm{E}^{AB}_{\alpha_i\beta_i} &= \sum_{i=1}^{m_1} \sqrt{\lambda^A_{\alpha_i}\lambda^B_{\beta_i}} (\mathrm{v}^A_{\alpha_i})^\top \mathrm{v}^B_{\beta_i} && (22)\\
&= \sum_{i=1}^{m_1}\sum_{j=1}^{m_2} \sqrt{\lambda^A_{\alpha_i}\lambda^B_{\beta_i}} v^A_{a_j\alpha_i} v^B_{b_j\beta_i} && (23)\\
&= \sum_{j=1}^{m_2}\sum_{i=1}^{m_1} r^A_{\alpha_i a_j} r^B_{\beta_i b_j} && (24)\\
&= \sum_{j=1}^{m_2} (\mathrm{r}^A_{a_j})^\top \mathrm{r}^B_{b_j} && (25)\\
&= \sum_{j=1}^{m_2} \mathrm{C}^{AB}_{a_jb_j}. && (26)
\end{aligned}$$

□

Although numerical experiments show that the contact alignment algorithm converges monotonically, it does not result in reliably good alignments. Presumably, this is because of well known difficulties of embedding contact maps in 3D [9], [31], [25]. Also, an incorrect orientation for an eigenspace could yield good contact space alignments but poor 3D alignments. In order to overcome these difficulties, we include 3D information in the alignment of the eigenspaces in step 3 of the algorithm.

For a given alignment of the protein chain, we superimpose the 3D coordinates of the folds using the widely used Kabsch procedure [17]. We then construct a 3D protein-protein contact matrix $\mathcal{C}_{AB} = [\mathcal{C}_{ij}^{AB}],\ i = 1, 2, \ldots, N_1,\ j = 1, 2, \ldots, N_2$, as follows. Assume X_A and X_B are the 3D coordinate of fold A and fold B. Then $\mathcal{C}_{ij}^{AB} = \sigma(d_{ij})$, where $\sigma(d_{ij})$ is the cutoff function defined in Figure 4 and $d_{ij} = \|\mathrm{X}_i^A - \mathrm{X}_j^B\|$ is the 3D distance between residue i in protein A and residue j in protein B. Finally, instead of using the standard inner product in step 3, we use $\mathrm{E}_{AB} = \tilde{R}_A \mathcal{C}_{AB} \tilde{R}_B^\top$.

The contact matrix $\mathcal{C}_{AB}$ serves as a 3D bridge between the contact spaces of folds A and B. Define the bilinear function $\langle \mathrm{u}, \mathrm{v} \rangle_{\mathcal{C}_{AB}} = \mathrm{u}^\top \mathcal{C}_{AB} \mathrm{v}$. The contact between residue i of fold A and residue j of fold B is $\langle \mathrm{e}_i^A, \mathrm{e}_j^B \rangle_{\mathcal{C}_{AB}} = \mathcal{C}_{ij}^{AB}$. The bilinear function uses linearity to extend this 3D contact information to the entire contact space of each of the folds. Protein-protein contacts can then be computed between the eigenvalue-weighted eigenvectors of each fold.

Unfortunately, the modification in step 3 of the algorithm does not preserve the monotonicity of the algorithm. We therefore run the modified algorithm for a fixed number of steps or until it converges, save the best alignment in terms of contact score (the score maximized in step 8) and then resort to alternate applications of the Kabsch procedure and DP to refine the alignment. We have observed that the contact alignment algorithm with the 3D modification quickly gives a good global alignment and the Kabsch procedure and DP quickly give a low RMSD alignment.

6 Hard Alignments

We report computational results for 10 difficult alignments studied in [7], [29]. We compare to four other alignment algorithms (data taken form [7]):

SAMO multi objective alignment algorithm that simultaneously minimizes RMSD and number of aligned residues [7].

Dali one of the first fold alignment algorithms. Dali uses distance matrix [15].

CE genetic algorithm based combinatorial extension algorithm [29].

LUND an algorithm that uses "fuzzy" contact matrices interpreted as probabilities [4].

The 3D version of our contact alignment algorithm followed by alternate applications of the Kabsch procedure and DP is denoted by EIGA (**EIG**ensystem **A**lignment). The spectrum algorithm is denoted by EIGAs. We use a cutoff of $\kappa = 8$ Angstroms for both EIGA and EIGAs, a gap penalty equal to zero for EIGA and one for EIGAs.

Tables 4 and 5 show that EIGA compares well to SAMO, Dali, CE and LUND, in terms of the number of aligned residues and RMSD error. Table 6 show that EIGA is reasonably fast. (We do not have specific computational times for the alignments of the other methods.) Additional alignment results are given in Tables 7–9 in the appendix.

Our earlier delta method [26] and the spectrum alignment algorithm (denoted EIGAs), do not compare favorably with the other methods for the hard alignments in Tables 4–5. But EIGAs does a reasonable job with the easier alignments and identifying different folds and fold classes as can be seen in Tables 7 and 8 in the appendix. In particular, EIGAs outperforms our earlier delta method in terms of both speed and quality. (The computing times reported in Table 6 and 9 include the preprocessing steps of each method. Alignment times will be even faster for database searches as the preprocessing step is done only once when a database is setup and is not required for searches.)

Figure 7 illustrates that despite poor RMSD values, EIGAs seems to do a good job of quickly forming a good global alignment. For 1FX1a vs 1UBQ, for example, EIGAs determines an alignment in 0.1 seconds. Figure 7(a) is the 3D protein-protein contact matrix of the superimposed structures. The RMSD is poor at 9.8. However, the global alignment is mostly correct with 74 aligned residues and a 3D contact score of 12.3. Applying the contact alignment algorithm (3D version) to EIGAs's alignment, reduces the number of aligned residues to 64 and RMSD to 5.2 and increases the 3D contact score to 24.6. Finally, a further application of 3D alignment, which involves alternating between DP and 3D superposition using the Kabsch proceedure, reduces the number of aligned residues to 63, RMSD to 2.6 and increases the 3D contact score to 45.4.

	Difficult Alignments		SAMO	Dali	CE	Lund	delta	EIGAs	EIGA
1.	1FXIa(96)	1UBQ (76)	70	60	64	63	59	74	63
2.	1TEN (195)	3HHRb(89)	87	86	87	87	75	88	86
3.	3HLAb(114)	2RHE (99)	87	75	84	83	73	95	82
4.	2AZAa(129)	1PAZ (120)	82	81	84	83	75	109	83
5.	1CEWi(108)	1MOLa(94)	83	81	81	82	65	88	81
6.	1CID (177)	2RHE (114)	98	97	97	97	101	113	98
7.	1CRL (534)	1EDE (310)	281	211	219	126	240	304	208
8.	2SIM (390)	1NSBa(381)	322	291	275	292	253	339	292
9.	1BGEb(159)	2GMFa(121)	110	94	107	104	96	121	101
10.	1TIE (166)	4FGF (124)	115	114	116	115	98	120	115

TABLE 4: Number of aligned residues.

	Difficult Alignments		SAMO	Dali	CE	Lund	delta	EIGAs	EIGA
1.	1FXIa(96)	1UBQ (76)	2.5	2.6	3.8	2.6	9.9	9.8	2.6
2.	1TEN (195)	3HHRb(89)	1.7	1.9	1.9	1.8	21.7	18.9	1.7
3.	3HLAb(114)	2RHE (99)	2.9	3.0	3.4	3.3	15.5	13.4	3.1
4.	2AZAa(129)	1PAZ (120)	2.5	2.5	2.9	2.4	15.4	10.9	2.4
5.	1CEWi(108)	1MOLa(94)	2.3	2.3	2.3	2.2	14.9	9.2	2.1
6.	1CID (177)	2RHE (114)	2.3	3.2	2.9	2.5	12.5	21.7	2.6
7.	1CRL (534)	1EDE (310)	3.1	3.5	3.8	5.0	24.8	23.6	3.1
8.	2SIM (390)	1NSBa(381)	2.9	3.3	3.0	2.9	21.5	17.0	3.0
9.	1BGEb(159)	2GMFa(121)	3.3	3.3	3.9	3.3	17.5	9.9	3.1
10.	1TIE (166)	4FGF (124)	2.4	3.1	2.9	2.7	16.3	13.5	2.7

TABLE 5: RMSD error (in Angstroms).

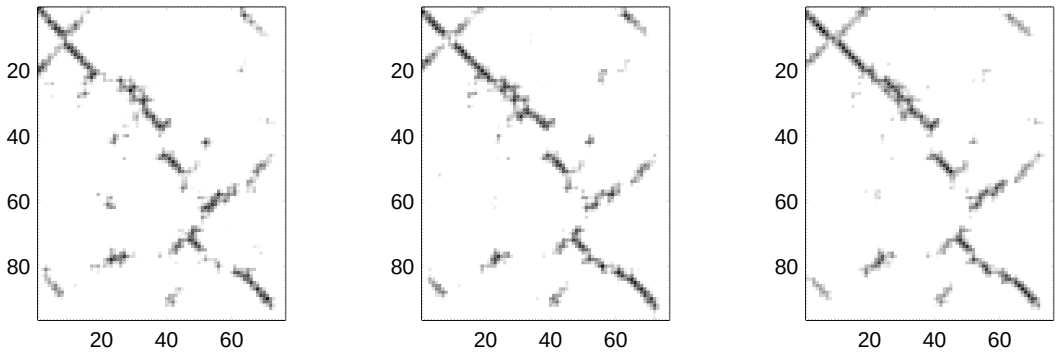
7 Conclusions

The contact geometry description of protein folds presented in this paper has a rich mathematical structure. We have used this mathematical structure to develop two new protein fold alignment algorithms, EIGAs and EIGA. Both are fast, but EIGAs is especially fast as it essentially runs at the speed of DP.

In a recent article, Hasegawa and Holm [13] claim that alignment methods that

	Difficult Alignments		delta	EIGAs	EIGA
1.	1FXIa(96)	1UBQ (76)	0.1	0.1	1.1
2.	1TEN (195)	3HHRb(89)	0.3	0.2	2.7
3.	3HLAb(114)	2RHE (99)	0.2	0.1	1.7
4.	2AZAa(129)	1PAZ (120)	0.2	0.1	3.7
5.	1CEWi(108)	1MOLa(94)	0.2	0.1	1.4
6.	1CID (177)	2RHE (114)	0.3	0.2	3.4
7.	1CRL (534)	1EDE (310)	3.4	2.5	29.1
8.	2SIM (390)	1NSBa(381)	2.6	1.8	24.3
9.	1BGEb(159)	2GMFa(121)	0.3	0.2	2.8
10.	1TIE (166)	4FGF (124)	0.3	0.2	6.8

TABLE 6: CPU time (in seconds).



(a) spectrum alignment (b) contact alignment (c) 3D alignment

Fig. 7: 1FX1a vs 1UBQ, protein-protein, 3D contact matrices after superposition using (a) spectrum alignment (b) refinement using contact alignment (3D version) and (c) final 3D alignment.

allow for flexibility generate the most biologically meaningful alignments. Instead of directly aligning the residues of a fold, the spectral alignment methods described in this paper first align the eigenspaces of the folds. Structural deformations observed in actual protein folds result in shifts in these eigenspaces. Since residues are referenced to their

eigenspaces, the final alignment naturally compensates for such structural deformations.

We are currently investigating a spectral approach to multiple structure alignment as well as refining the mathematical description of the 3D version of our contact alignment algorithm. We are also working on a detailed mathematical analysis of the alignment error of protein alignments.

ACKNOWLEDGMENTS

The idea of using the eigensystem of protein contact maps to align protein structures was inspired by the work of Galaktionov and Marshall [12] on protein structure prediction. Preliminary ideas for the algorithms 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] 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.

[2] 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.

[3] 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.

[4] Richard Blankenbecler, Mattias Ohlsson, Carsten Peterson, and Markus Ringner. Matching protein structures with fuzzy alignments. *Proc Natl Acad Sci U S A*, 100(21):11936–11940, Oct 2003.

[5] Forbes J. Burkowski. *Structural Bioinformatics an Algorithmic Approach*. CRC Press, 2009.

[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] Luonan Chen, Ling-Yun Wu, Yong Wang, Shihua Zhang, and Xiang-Sun Zhang. Revealing divergent evolution, identifying circular permutations and detecting active-sites by protein structure comparison. *BMC Struct Biol*, 6:18, 2006.

[8] 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.

[9] G. M. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformations*. Wiley, 1988.

[10] Ingvar Eidhammer, Inge Jonassen, and William Taylor. *Protein Bioinformatics: An Algorithmic Approach to Sequence and Structure Analysis*. John Wiley and Sons, 2004.

[11] R. J. Forrester and H. J. Greenberg. Quadratic binary programming models in computational biology. *Algorithmic Operations Research*, 3:110–129, 2008.

[12] 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.

[13] Hitomi Hasegawa and Liisa Holm. Advances and pitfalls of protein structural alignment. *Curr Opin Struct Biol*, 19(3):341–348, Jun 2009.

[14] 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.

[15] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J Mol Biol*, 233(1):123–138, Sep 1993.

[16] Neil C. Jones and Pavel A. Pevzner. *An Introduction to Bioinformatics Algorithms*. MIT Press, 2004.

[17] W Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallog A*, 34:827–828, 1978.

[18] 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.

[19] G. Lancia, R. Carr, B. Walenz, and 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] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–453, Mar 1970.

[22] Ben Noble and James W. Daniel. *Applied Linear Algebra*. Prentice-Hall, 1977.

[23] 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.

[24] Aleksandar Poleksic. Algorithms for optimal protein structure alignment. *Bioinformatics*, 25(21):2751–2756, Nov 2009.

[25] S. Saitoh, T. Nakai, and K. Nishikawa. A geometrical constraint approach for reproducing the native backbone conformation of a protein. *Proteins*, 15(2):191–204, Feb 1993.

[26] Y. Shibberu, A. Holder, and K. Lutz. Fast protein structure alignment. In M. Borodovsky, editor, *LNCS (LNBI)*, volume 6053, pages 152–165, Berlin, 2010. Springer-Verlag.

[27] Tetsuo Shibuya. Fast hinge detection algorithms for flexible protein structures. *IEEE/ACM Trans Comput Biol Bioinform*, 7(2):333–341, 2010.

[28] Tetsuo Shibuya, Jesper Jansson, and Kunihiko Sadakane. Linear-time protein 3-d structure searching with insertions and deletions. *Algorithms Mol Biol*, 5:7, 2010.

[29] I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein Eng*, 11(9):739–747, Sep 1998.

[30] 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.

[31] M. Vendruscolo, E. Kussell, and E. Domany. Recovery of protein structure from contact maps. *Fold Des*, 2(5):295–306, 1997.

[32] Yong Wang, Ling-Yun Wu, Ji-Hong Zhang, Zhong-Wei Zhan, Xiang-Sun Zhang, and Luonan Chen. Evaluating protein similarity from coarse structures. *IEEE/ACM Trans Comput Biol Bioinform*, 6(4):583–593, 2009.

APPENDIX A

		SAMO	Dali	CE	Lund	delta	EIGAs	EIGA
Reductases								
1DHFa(186)	8DFR (182)	182	182	182	182	175	177	182
1DHFa(182)	4DFRa(159)	153	154	154	156	122	156	155
1DHFa(182)	3DFR (162)	159	158	158	159	137	158	159
8DFR (186)	4DFRa(159)	157	155	155	157	126	157	156
8DFR (186)	3DFR (162)	159	159	158	160	129	160	160
4DFRa(162)	3DFR (159)	156	154	155	155	129	153	155
Globins								
2HHBa(146)	2HHBb(141)	139	138	139	139	102	136	139
2HHBa(153)	1MBD (141)	141	139	141	141	113	138	141
2HHBa(147)	2HBG (141)	140	138	136	138	107	132	138
2HHBa(141)	1ECD (136)	131	129	128	130	128	131	130
1MBD (153)	2HBG (147)	141	140	140	140	85	141	139
2HHBb(153)	1MBD (146)	145	145	144	145	130	142	145
2HHBb(147)	2HBG (146)	137	135	134	136	95	134	136
2HHBb(146)	1MBA (146)	140	138	139	140	110	137	141
2LHB (153)	1MBD (149)	137	135	137	137	111	135	137
2LHB (149)	2HBG (147)	133	128	130	132	106	132	131
1MBD (153)	1MBA (146)	143	142	141	143	100	138	143
1MBA (146)	1ECD (136)	136	133	134	136	93	131	136
2HBG (147)	1ECD (136)	129	129	125	129	120	129	129
Different Folds								
1NSBa(390)	1TIE (166)	156	-	88	-	140	166	81
1NSBa(390)	4FGF (124)	118	-	72	-	110	124	76
1FXIa(108)	1CEWi(96)	56	-	56	-	75	90	42
1FXIa(96)	1MOLa(94)	70	-	48	-	66	84	48
Different Classes								
1BGEb(159)	1TEN (89)	82	-	40	-	66	86	44
1BGEb(159)	1PAZ (120)	103	-	48	-	105	113	50
2GMFa(121)	1TEN (89)	68	-	40	-	72	87	46

TABLE 7: Number of aligned residues.

		SAMO	Dali	CE	Lund	delta	EIGAs	EIGA
Reductases								
1DHFa(186)	8DFR (182)	0.7	0.7	0.7	0.7	1.4	2.3	0.7
1DHFa(182)	4DFRa(159)	1.8	2.0	2.0	2.0	10.5	3.8	1.9
1DHFa(182)	3DFR (162)	1.6	1.7	1.7	1.7	6.8	5.0	1.7
8DFR (186)	4DFRa(159)	1.9	2.0	2.0	2.0	8.4	6.1	1.9
8DFR (186)	3DFR (162)	1.6	1.8	1.8	1.8	9.6	4.3	1.8
4DFRa(162)	3DFR (159)	1.5	1.5	1.5	1.5	5.8	3.5	1.5
Globins								
2HHBa(146)	2HHBb(141)	1.4	1.4	1.5	1.4	15.5	4.6	1.4
2HHBa(153)	1MBD (141)	1.5	1.5	1.6	1.5	13.7	3.5	1.5
2HHBa(147)	2HBG (141)	1.6	1.7	1.7	1.6	12.8	4.2	1.6
2HHBa(141)	1ECD (136)	2.2	2.3	2.6	2.3	8.7	5.8	2.3
1MBD (153)	2HBG (147)	2.0	2.2	2.1	2.0	18.0	4.3	1.9
2HHBb(153)	1MBD (146)	1.6	1.6	1.6	1.6	5.3	3.0	1.6
2HHBb(147)	2HBG (146)	1.7	2.0	1.9	1.7	18.1	4.9	1.7
2HHBb(146)	1MBA (146)	2.2	2.3	2.4	2.4	7.9	5.0	2.4
2LHB (153)	1MBD (149)	1.4	1.4	1.6	1.5	14.5	4.7	1.5
2LHB (149)	2HBG (147)	1.9	2.0	2.1	2.0	15.2	5.8	1.9
1MBD (153)	1MBA (146)	1.9	1.9	1.8	1.9	14.2	4.5	1.9
1MBA (146)	1ECD (136)	1.9	1.9	2.0	2.0	12.6	4.4	1.9
2HBG (147)	1ECD (136)	2.4	2.6	2.6	2.5	7.8	7.1	2.5
Different Folds								
1NSBa(390)	1TIE (166)	3.1	-	6.4	-	20.9	20.8	4.2
1NSBa(390)	4FGF (124)	3.0	-	5.8	-	15.1	18.2	4.5
1FXIa(108)	1CEWi(96)	2.9	-	7.2	-	15.2	16.0	3.7
1FXIa(96)	1MOLa(94)	2.9	-	5.8	-	14.2	13.7	3.7
Different Classes								
1BGEb(159)	1TEN (89)	2.8	-	7.4	-	15.7	16.0	3.9
1BGEb(159)	1PAZ (120)	3.2	-	6.2	-	16.6	18.5	4.1
2GMFa(121)	1TEN (89)	3.0	-	4.8	-	11.9	15.4	3.7

TABLE 8: RMSD error in Angstroms.

		delta	EIGAs	EIGA
Reductases				
1DHFa(186)	8DFR (182)	0.6	0.4	2.5
1DHFa(182)	4DFRa(159)	0.4	0.3	3.0
1DHFa(182)	3DFR (162)	0.4	0.3	2.4
8DFR (186)	4DFRa(159)	0.4	0.3	3.6
8DFR (186)	3DFR (162)	0.4	0.3	3.1
4DFRa(162)	3DFR (159)	0.4	0.2	2.1
Globins				
2HHBa(146)	2HHBb(141)	0.3	0.2	2.1
2HHBa(153)	1MBD (141)	0.3	0.2	1.8
2HHBa(147)	2HBG (141)	0.3	0.2	1.9
2HHBa(141)	1ECD (136)	0.3	0.2	1.8
1MBD (153)	2HBG (147)	0.3	0.2	2.5
2HHBb(153)	1MBD (146)	0.3	0.2	2.1
2HHBb(147)	2HBG (146)	0.3	0.2	6.9
2HHBb(146)	1MBA (146)	0.3	0.2	1.7
2LHB (153)	1MBD (149)	0.3	0.2	7.2
2LHB (149)	2HBG (147)	0.3	0.2	2.8
1MBD (153)	1MBA (146)	0.3	0.2	2.3
1MBA (146)	1ECD (136)	0.3	0.2	2.0
2HBG (147)	1ECD (136)	0.3	0.2	2.0
Different Folds				
1NSBa(390)	1TIE (166)	1.3	0.9	32.1
1NSBa(390)	4FGF (124)	1.1	0.8	19.6
1FXIa(108)	1CEWi(96)	0.2	0.1	3.8
1FXIa(96)	1MOLa(94)	0.1	0.1	1.9
Different Classes				
1BGEb(159)	1TEN (89)	0.2	0.2	2.8
1BGEb(159)	1PAZ (120)	0.3	0.2	6.5
2GMFa(121)	1TEN (89)	0.2	0.1	3.0

TABLE 9: CPU time (in seconds).