MODELING AND SUPERPOSITION OF MULTIPLE PROTEIN STRUCTURES USING AFFINE TRANSFORMATIONS: ANALYSIS OF THE GLOBINS

THOMAS D. WU[1], SCOTT C. SCHMIDLER[1,3], TREVOR HASTIE[2], DOUGLAS L. BRUTLAG[1]

Departments of [1] Biochemistry and [2] Statistics and [3] Section on Medical Informatics
Stanford University, Stanford, CA 94305, USA

A novel approach for analyzing multiple protein structures is presented. A family of related protein structures may be characterized by an affine model, obtained by applying transformation matrices that permit both rotation and shear. The affine model and transformation matrices can be computed efficiently using a single eigendecomposition. A novel method for finding correspondences is also introduced. This method matches curvatures along the protein backbone. The algorithm is applied to analyze a set of seven globin structures. Our method identifies 100 corresponding landmarks across all seven structures. Results show that most helices in globins can be identified by high curvature, with the exception of the C and D helices. Analysis of the superposition reveals that globins are most strongly conserved structurally in the mid-regions of the E and G helices.

1 Introduction

Analysis of protein structures is most informative when we can examine a family of related structures, rather than a single structure. Families of structures are more revealing because commonalities and variations within the family suggest the relative importance and possible functions of amino acid residues. Protein structural families are generally studied by rotating the structures to superimpose corresponding atoms. From this multiple superposition, researchers sometimes calculate an "average" structure by averaging the coordinates of the corresponding atoms.[1]

In this paper, we take a different approach to analyzing multiple protein structures. We assume that variations in protein structure can be represented by a statistical model. We solve that model to obtain a "template" structure that describes the entire family. Our statistical model allows each protein structure to undergo a general affine transformation, which permits not only rotations, as in existing approaches, but also shear in three dimensions.

Because we allow affine transformations, we can perform averaging and superposition simultaneously. Our method essentially finds the eigensolution to a least-squares problem. In the least-squares formulation, we find an affine model $\overline{\mathbf{M}}$ and transformation matrices $\mathbf{B}_j$ that minimize the objective function

$$\sum_{j=1}^{J} ||\mathbf{M}_j\mathbf{B}_j - \overline{\mathbf{M}}||^2 \qquad (1)$$

where each $\mathbf{M}_j$ is a coordinate matrix of corresponding atoms in the J protein structures. It can be shown that, under certain assumptions, the model found is the maximum likelihood estimate of the average protein structure.[2] Therefore, we view the analysis of protein families not merely as a superposition problem, but as a problem in modeling. The modeling approach is more general than superposition, and can be extended to handle different assumptions about how protein structures vary.

Given a set of correspondences, we can obtain the affine model quickly, using a single eigendecomposition.[3] In contrast, when only rotations are allowed, the optimal superposition of multiple structures requires iterative or stochastic techniques.[4,5,6,7] Existing methods try to find rotations $\mathbf{\Gamma}_j$ that minimize the sum of distances between all pairs of structures:

$$\sum_{i=1}^{J} \sum_{j=i+1}^{J} ||\mathbf{M}_i\mathbf{\Gamma}_i - \mathbf{M}_j\mathbf{\Gamma}_j||^2$$

This rotational superposition requires an iterative algorithm, whereas the optimal affine model can be obtained in a single step.

Both the rotational and the affine approaches require a correspondence among atoms in the different protein structures. In this paper, we also present a novel method for the correspondence problem. We show that an initial correspondence can be found quickly and relatively accurately by matching curvatures. We have found that curvature along the backbone of a protein structure serves as a useful "signature" that highlights the main features of the structure, including helices, strands, and loops. Moreover, curvature is invariant to rotation and location. Therefore, curvature allows us to make a preliminary correspondence among protein structures, without having to superimpose the structures beforehand. The method of matching curvatures provides an alternative to existing methods for finding correspondences. Because curvature is a scalar function, it simplifies the dynamic programming procedure compared with matching vector sets,[8,9] distance matrices,[10] or properties.[11]

In the remainder of the paper, we present procedures for computing curvatures, affine models, and affine superpositions. These procedures constitute a complete algorithm for modeling and superimposing multiple protein structures. We then apply our method to analyze the globin family, and identify the commonalities and variations that are revealed using affine transformations.

2 Methods

2.1 *Formulation*

The affine transformation algorithm takes as input a set of J protein structures, with arbitrary orientations and shifts relative to the origin. Each protein structure is represented by a *structure matrix* $\mathbf{S}_j$, which is an $N_j \times 3$ matrix of coordinates. Each row in the matrix contains the x, y, and z coordinates for an atom in the protein structure. In this paper, we consider only the backbone, represented by the Cα atoms, ordered from the N- to the C-terminus. However, it is possible to extend our approach to consider other atoms as well.

We adopt a statistical model for the observed protein structures. We assume that the given family may be described by a template or *affine model* $\overline{\mathbf{M}}$, which is an $N \times 3$ matrix of landmarks. A *landmark* is a point that is assumed to be common to all structures in the given family. A set of landmarks describes a *correspondence* among the protein structures. The corresponding landmarks for each protein structure $\mathbf{S}_j$ are represented by a *landmark matrix* $\mathbf{M}_j$, which contains a subset of the atoms in $\mathbf{S}_j$. Our model assumes that each landmark matrix $\mathbf{M}_j$ differs from the affine model $\overline{\mathbf{M}}$, both globally and locally, by the following relationship:

$$\mathbf{M}_j = \overline{\mathbf{M}}\mathbf{B}_j^{-1} + \mathbf{1}\boldsymbol{\mu}_j^T + \boldsymbol{\varepsilon}_j \tag{2}$$

where $\mathbf{B}_j$ is a 3×3 affine *transformation matrix*, $\boldsymbol{\mu}_j$ is a 3×1 *offset vector*, and $\boldsymbol{\varepsilon}_j$ is an $N \times 3$ *error matrix*. The offset vector describes the global translation difference between each structure and the affine model; the transformation matrix describes the global rotational and shear difference; and the error matrix represents local differences for each landmark.

Our algorithm consists of three steps: (1) Compute a curvature function κ_j for each protein structure $\mathbf{S}_j$. Find corresponding landmarks $\mathbf{M}_j^{(1)}$ by *matching curvatures to a reference structure*, and obtain the affine model $\overline{\mathbf{M}}^{(1)}$ and transformation matrices $\mathbf{B}_j^{(1)}$. (2) Find corresponding landmarks $\mathbf{M}_j^{(2)}$ by *matching coordinates to a reference structure*, and obtain the affine model $\overline{\mathbf{M}}^{(2)}$ and transformation matrices $\mathbf{B}_j^{(2)}$. (3) Find corresponding landmarks $\mathbf{M}_j$, by *matching coordinates iteratively to the evolving affine model*, and obtain the affine model $\overline{\mathbf{M}}$ and transformation matrices $\mathbf{B}_j$.

Our algorithm spends most of its time finding and refining corresponding landmarks. If the landmarks were known in advance, our method would require only a single step to obtain the affine model and transformation matrices.

2.2 Curvatures

Our algorithm begins by computing the curvature at each Cα carbon along the backbone of each protein structure. Let us label each Cα atom in protein $\mathbf{S}_j$ by a *sequence index* $s = 1, \ldots, N_j$. Since the series of peptide bond lengths between adjacent Cα carbons has relatively constant length, the index s also serves as an arc length parameter along the backbone. Let the position of Cα at index s be $\mathbf{p}_j^T(s) = [x_j(s)\ y_j(s)\ z_j(s)]$, where $x_j(s)$, $y_j(s)$, and $z_j(s)$ are coordinates from structure matrix $\mathbf{S}_j$.

Then we perform two rounds of numerical differentiation:

$$\begin{aligned} \Delta\mathbf{p}_j(s) &= [\mathbf{p}_j(s+1) - \mathbf{p}_j(s-1)]/2 & (3) \\ \mathbf{t}_j(s) &= \frac{\Delta\mathbf{p}_j(s)}{||\Delta\mathbf{p}_j(s)||} & (4) \\ \frac{d\mathbf{t}_j(s)}{ds} &= [\mathbf{t}_j(s+1) - \mathbf{t}_j(s-1)]/2 & (5) \\ \kappa_j(s) &= \left\| \frac{d\mathbf{t}_j(s)}{ds} \right\| & (6) \end{aligned}$$

where $\mathbf{t}_j(s)$ is the unit tangent vector and $\kappa_j(s)$ is the curvature at $s = 3, \ldots, N_j - 2$.

2.3 Corresponding Landmarks

We compute corresponding landmarks using dynamic programming, which is also known as dynamic time-warping in other literature.[12] Dynamic programming finds a correspondence between two structures that minimizes the overall distance between the structures. Let r and s be the sequence indices of atoms in structure matrices $\mathbf{S}_i$ and $\mathbf{S}_j$, respectively. Let $d(r,s)$ be some distance metric between atoms r and s. Then we would like to find two collinear sequences of atoms $1 \le r_{(1)} < r_{(2)} < \ldots < r_{(m)} \le N_r$ and $1 \le s_{(1)} < s_{(2)} < \ldots < s_{(m)} \le N_s$ that minimize the function

$$\begin{aligned} \sum_{i=1}^{m} d(r_{(i)}, s_{(i)}) + g(0, r_{(1)}) + \sum_{i=1}^{m-1} h(r_{(i)}, r_{(i+1)}) + g(r_{(m)}, N_r + 1) \\ + g(0, s_{(1)}) + \sum_{i=1}^{m-1} h(s_{(i)}, s_{(i+1)}) + g(s_{(m)}, N_s + 1) \qquad (7) \end{aligned}$$

where $g(x,y)$ is the *gap penalty* for skipping from x to y at the end of either sequence, and $h(x,y)$ is the gap penalty for skipping from x to y in the middle of either sequence. Gap penalty functions may be arbitrarily complex, but when they are linear functions, the time complexity is reduced from $O(n^3)$ to $O(n^2)$.[13] Hence, we use $g(x,y) = \alpha_0 + \beta_0(y-x)$, and $h(x,y) = \alpha_1 + \beta_1(y-x)$, except we require the gap penalty be zero when $y - x = 1$. The parameters α are opening penalties, and the parameters β are extension penalties. We typically choose smaller penalties for g, because protein sequences often have variable-length ends that do not correspond well to other sequences.

The three steps apply dynamic programming with different distance metrics:

$$d(r,s) = \begin{cases} (\kappa_i(r) - \kappa_j(s))^2 & \text{for step 1} \\ ||\mathbf{p}_i(r) - \mathbf{p}_j(s)||^2 & \text{for step 2} \\ ||\mathbf{p}_{\overline{\mathbf{M}}'}(r) - \mathbf{p}_j(s)||^2 & \text{for step 3} \end{cases} \tag{8}$$

In steps 1 and 2, we compute distances relative to a reference structure $\mathbf{S}_i$. The reference for step 1 is simply the longest protein. The reference for step 2 is the protein structure closest to the initial affine model obtained in step 1. In step 3, we transform the affine model $\overline{\mathbf{M}}$ into a structure $\overline{\mathbf{M}}'$ in the space of $\mathbf{S}_j$, and then measure distances relative to the transformed model. We explain this transformation further in Section 2.5.

2.4 Affine Models and Transformation Matrices

For each round of corresponding landmarks, our algorithm computes an affine model and set of transformation matrices. Let us assume that each landmark matrix $\mathbf{M}_j$ is centered by subtracting $\mathbf{1}\boldsymbol{\mu}_j$, where the offset vector $\boldsymbol{\mu}_j$ contains the mean x, y, and z coordinates of $\mathbf{M}_j$. Our algorithm stores $\boldsymbol{\mu}_j$ at each step, for use in later superpositions.

The objective function in equation 1 can be solved using least-squares regression, following a method described by Hastie and colleagues.[3] To avoid degeneracies, we require that $\overline{\mathbf{M}}$ is orthogonal, so $\overline{\mathbf{M}}^T\overline{\mathbf{M}} = \mathbf{I}$. Then the optimal transformation matrix for $\mathbf{M}_j$ is $\mathbf{B}_j = (\mathbf{M}_j^T\mathbf{M}_j)^{-1}\mathbf{M}_j^T\overline{\mathbf{M}}$ and hence $\mathbf{M}_j\mathbf{B}_j = \mathbf{H}_j\overline{\mathbf{M}}$, where $\mathbf{H}_j = \mathbf{M}_j(\mathbf{M}_j^T\mathbf{M}_j)^{-1}\mathbf{M}_j^T$ is a projection operator. So at the minimum, the quantity in equation 1 equals

$$\sum_{j=1}^{J} ||(\mathbf{H}_j - \mathbf{I})\overline{\mathbf{M}}||^2 = \sum_{j=1}^{J} \mathrm{tr}(\overline{\mathbf{M}}^T(\mathbf{I} - \mathbf{H}_j)\overline{\mathbf{M}}) = J\mathrm{tr}(\overline{\mathbf{M}}^T(\mathbf{I} - \overline{\mathbf{H}})\overline{\mathbf{M}}) \tag{9}$$

where $\overline{\mathbf{H}}$ is the average of the projection operators $\mathbf{H}_j$. The solution for the affine model that minimizes the above quantity can be obtained by letting $\overline{\mathbf{M}}$ be the eigenvectors corresponding to the three largest eigenvalues of $\overline{\mathbf{H}}$.

In practice, to achieve better numerical stability, we perform the above computations by using QR decompositions $\mathbf{M}_j = \mathbf{Q}_j\mathbf{R}_j$, where $\mathbf{Q}_j$ is orthogonal and $\mathbf{R}_j$ is upper triangular. This decomposition then allows us to compute $\mathbf{H}_j = \mathbf{Q}_j\mathbf{Q}_j^T$ and $\mathbf{B}_j = \mathbf{R}_j^{-1}\mathbf{Q}_j^T\overline{\mathbf{M}}$.

2.5 Affine Superpositions

Several steps of our algorithm compare one protein structure to another or to the affine model. We make these comparisons using the transformation matrices and offset vectors computed in the previous section. To superimpose the affine model $\overline{\mathbf{M}}$ onto a protein structure $\mathbf{S}_j$ or landmark matrix $\mathbf{M}_j$, we apply the inverse of the transformation matrix to obtain

$$\overline{\mathbf{M}}' = \overline{\mathbf{M}}\mathbf{B}_j^{-1} + \mathbf{1}\boldsymbol{\mu}_j^T \tag{10}$$

where $\boldsymbol{\mu}_j$ is the offset vector for $\mathbf{M}_j$.

To superimpose one protein structure $\mathbf{S}_i$ onto another $\mathbf{S}_j$, we transform $\mathbf{S}_i$ into the model space and then transform it into the space of $\mathbf{S}_j$:

$$\mathbf{S}_i' = (\mathbf{S}_i - \mathbf{1}\boldsymbol{\mu}_i^T)\mathbf{B}_i\mathbf{B}_j^{-1} + \mathbf{1}\boldsymbol{\mu}_j^T \tag{11}$$

where $\mathbf{B}_i$ and $\mathbf{B}_j$ are transformation matrices and $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ are offset vectors for $\mathbf{M}_i$ and $\mathbf{M}_j$, respectively.

2.6 Decomposition of Transformations

The affine superpositions described in the previous section may introduce shear components. The amount of shear may be determined as follows. Let us consider the superposition matrix $\mathbf{T} = \mathbf{B}_i\mathbf{B}_j$ that superimposes structure $\mathbf{S}_i$ onto $\mathbf{S}_j$. This matrix can be decomposed into a pure rotation $\mathbf{R}$ followed by a pure scaling $\mathbf{D}$, then a shear $\mathbf{Z}$:

$$\mathbf{T} = \mathbf{RDZ} \tag{12}$$

where $\mathbf{R}$ is orthogonal, $\mathbf{D}$ is diagonal, and $\mathbf{Z}$ is upper triangular with ones along the diagonal. We can solve for the components by letting $\mathbf{G}$ be the Cholesky decomposition of $\mathbf{T}^T\mathbf{T}$ and setting $\mathbf{D}$ equal to the diagonal entries of $\mathbf{G}$. Then $\mathbf{Z} = \mathbf{D}^{-1}\mathbf{G}$ and $\mathbf{R} = \mathbf{TG}^{-1}$. The upper triangular entries of $\mathbf{Z}$ measure shear of each axis relative to other axes.

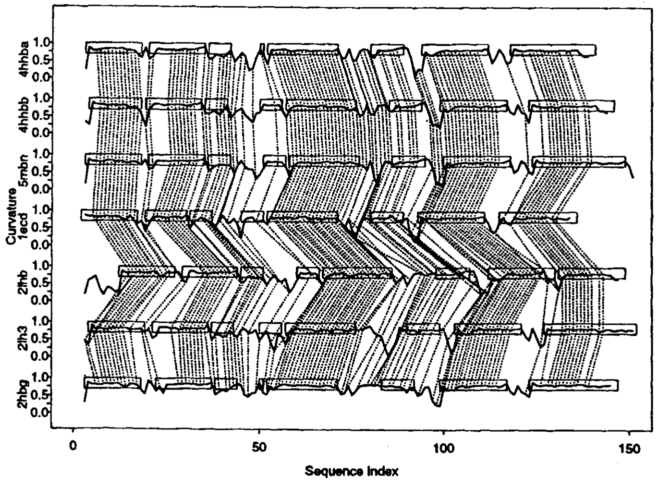

Figure 1: Curvature and curvature-based correspondence for globins. The curvature function for each globin is drawn in heavy lines. The location of helices for each globin are denoted by rectangles. Corresponding landmarks are shown as dotted lines between curvature functions.

3 Results

We now present a case study involving the globin family, chosen largely because it has been studied extensively in prior studies of protein structure families.[14,8,1] We studied the seven globin structures examined by Bashford and colleagues:[14] human deoxy-hemoglobin α (PDB accession 4HHBA) and β (4HHBB), sperm whale deoxy-myoglobin (5MBN), larval deoxy-hemoglobin (from *Chironomous thummi*, 1ECD), sea lamprey cyano-hemoglobin (2LHB), yellow lupin root nodule cyano-leghemoglobin (*Lupinus luteus*, 2LH3), and annelid worm deoxy-hemoglobin (*Glycera dibranchiata*, 2HBG).

We implemented the algorithm in the statistical computing language S-Plus, except for the pairwise dynamic programming procedure, which was written in C and loaded dynamically into S-Plus. We executed the program on a Silicon Graphics O2 workstation with a 175 MHz MIPS R10000 processor. For gap penalties, we set all opening penalties α to 0. The remaining penalties were: $\beta_0 = 0.01$, $\beta_1 = 0.02$, in step 1; $\beta_0 = 8$, $\beta_1 = 16$, in step 2; and $\beta_0 = 4$, $\beta_1 = 8$, in step 3. Our algorithm required 12 CPU seconds to execute.

The curvature functions are shown in Figure 1. The figure also shows the location of the α-helices for each globin, as defined by Bashford and colleagues.

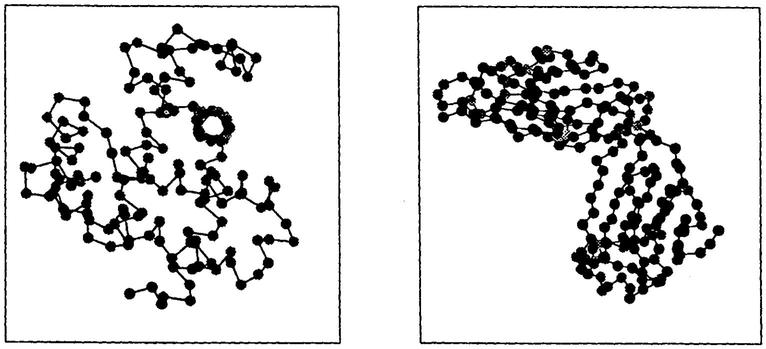

Figure 2: Relationship between curvature and secondary structure. The figure on the left is sperm whale myoglobin (5MBN); on the right is the variable light chain of immunoglobulin NC41 (1NCA). Each Cα carbon is shaded according to its curvature, from black (curvature of 0.0) to white (1.0).

Each curvature function has discrete regions of relatively high and constant curvature—corresponding to the helices—separated by regions of lower, more variable curvature—corresponding to the loops. The curvature is unusually variable in the C and D helices, and only 4HHBB, 5MBN, and 2LHB have clearly defined D helices. Between helices F and G, all structures have a sequence of 2 to 3 amino acids where the curvature drops sharply. The common element to all sequences appears to be a small hydrophobic residue—valine or isoleucine—surrounded by one or more charged amino acids.

The relationship between curvature and secondary structure is illustrated clearly in Figure 2, which maps curvature onto the three-dimensional structures of a globin and an immunoglobulin. The high curvature for α-helices in globins contrasts sharply with the low curvature for β-strands in immunoglobulins. Loops generally have intermediate curvature.

In step 1, our algorithm found 104 corresponding landmarks by matching curvatures to the reference structure 5MBN, chosen because it is longest. The pairwise dynamic programming procedure for matching curvature is demonstrated in Figure 3 (left), which shows the match between 5MBN and 1ECD. The resulting set of landmarks for all structures is shown as dashed lines in Figure 1. Landmarks were found primarily in the helices, and virtually none in the regions between helices.

In step 2, our algorithm found 108 corresponding landmarks by matching coordinates to the reference structure 5MBN, chosen because it was closest to the affine model obtained in step 1. The coordinate-based pairwise dynamic programming procedure is illustrated in Figure 3 (right).

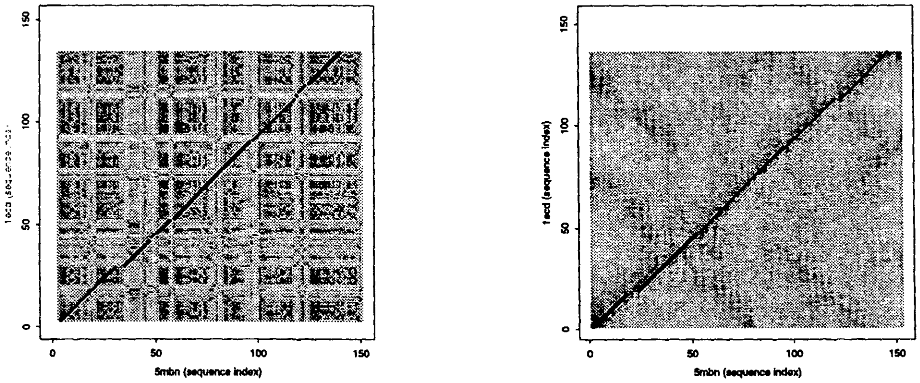

Figure 3: Dynamic programming method for finding correspondences, using curvature (left) and coordinate distance (right). Distance metrics are plotted as images, with darker intensity representing smaller distance. Optimal solutions are plotted as points on each graph.

Step 3 of the algorithm refined the sets of landmarks in four iterations, yielding 102, 100, 100, and 100 landmarks, successively.

For comparison, the manual model by Bashford and colleagues has 115 positions that could be considered landmarks (i.e., having one representative from each globin). Their landmarks generally agreed with ours in the B, C, E, and G helices. Differences occurred primarily in the A helix, where our model had 2LH3 shifted by 3 to 4 residues relative to the manual model, and in the F helix, where our model had 2LH3 shifted by one residue. The manual model has no landmarks in the D helix, whereas our model has one landmark there. Our model also omitted the landmarks in the extreme N and C termini of the globin sequences. The final correspondence is listed in Figure 5.

Our final superposition is shown in Figure 4, which shows the affine model and each of the original protein structures superimposed onto the space of 5MBN.

The amount of structural variation can be quantified by measuring residual deviations from the model. We compute the residuals by superimposing the affine model onto each landmark matrix and measuring the differences between corresponding coordinates. The resulting residual matrix $\mathbf{e}_j$ provides an estimate of the error matrix ε_j in equation 2. If we assume that ε_j is isotropic, the unbiased estimate of the variance at each landmark s is the mean square error $\sum_{j=1}^{J} ||\mathbf{e}_j(s)||^2/(J-1)$. Figure 5 shows the variability across all land-

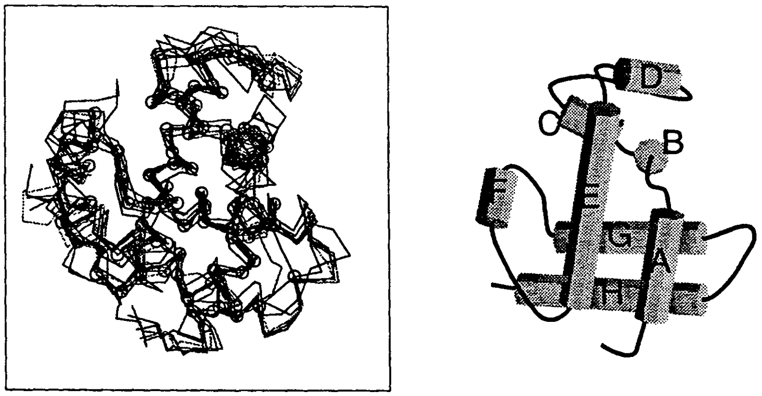

Figure 4: Affine model and superposition of globins. The superposition is shown on the left, with a schematic on the right. The affine model is represented by a chain of open circles. The seven globins are represented by line segments.

marks. We see that the mid-regions of helices E and G are conserved the most, which makes physiologic sense, since both helices make close contact with the heme group. The distal histidine residue in the E helix, which interacts with the heme group, is particularly well conserved.

To determine the amount of shear introduced, we decomposed the transformation matrices as in section 2.6, using transformations onto the space of 5MBN. The average shear component was 1.1% with a standard deviation of 4.0%. Shear ranged from -9.2% (in one component of 1ECD) to $+8.7\%$ (in one component of 2LH3). Nevertheless, the affine model showed reasonable geometry. For comparison, we computed a purely rotational model using the same landmarks. The root mean square difference between Cα-Cα bond lengths in the two models was 0.09Å, and between Cα-Cα bond angles, it was only 1.9°.

4 Discussion

We have developed a new approach to analyzing families of protein structures. Our approach introduces a different way of thinking about the problem. The problem changes from a superposition task to a modeling task, and the algorithm changes from finding rotations to finding affine models.

Our approach is closely related to regression analysis.[2] This relationship opens the possibilities for new ways to analyze protein families, especially because the field of regression analysis is well developed. The speed and simplicity of our approach also creates new opportunities for further study. Our method

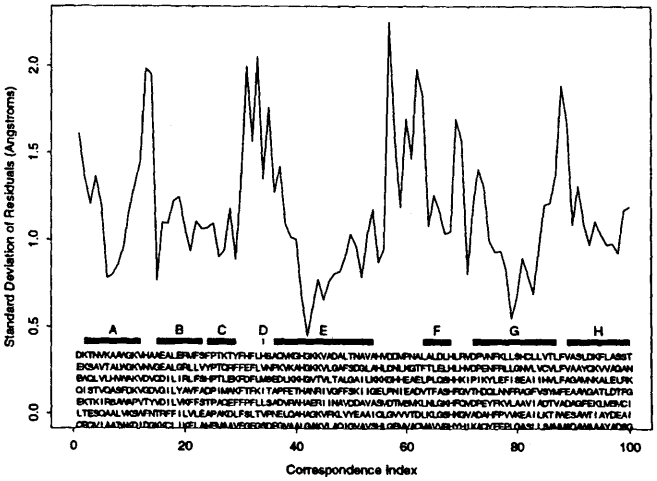

Figure 5: Structural variability of the globins. Residual standard deviations are plotted versus correspondence index. Helices are marked by solid rectangles and labeled from A to H. The corresponding amino acids for the seven globins are also printed.

generates affine models in a matter of seconds, and this speed may permit other types of investigations, such as cluster analyses of protein structures.

By allowing shear, we introduce some new issues in protein modeling. Although our method may appear to allow unnatural amounts of shear, the resulting affine model is constrained by the matching bond lengths and angles in the data. In the case of the globins, the affine model had reasonable geometry in comparison with a purely rotational model. Second, shear adds flexibility to the comparison of different structures. This allows us to see similarities between protein structures that more constrained methods may miss. For instance, by relaxing the rotational constraint, Diamond[15] was able perform a pairwise superposition of oxy- and deoxyhemoglobin. Protein structures are flexible, rather than rigid objects.

Curvature matching is similar to hierarchical secondary structure methods for finding correspondences,[16] but does not require prior definitions of secondary structure. Moreover, curvature analysis may be useful in other applications. Curvature reveals secondary structure elements readily, and matches of curvature may show quickly whether two structures are similar, even without performing a superposition. Hence, curvature may be useful for scanning the structural database quickly.

Studies of curvature may also provide insights into protein structure. Currently, local conformations of amino acids are characterized by ϕ-ψ angles, which represent torsional angles between adjacent residues. Because curvature represents local conformation differently, further studies of curvature may enhance our understanding of the sequence-structure relationship in proteins.

Acknowledgments

TDW is a Howard Hughes Medical Institute Physician Postdoctoral Fellow. SCS is supported by NLM training grant LM-07033. TH is supported by NSF grant DMS-9504495 and NIH grant ROI-CA-72028. DLB is supported by NLM grant LM-05716.

References

1. Mark Gerstein and Russ B. Altman. *CABIOS*, 11:633–644, 1995.
2. Colin Goodall. *J. Royal Stat. Soc. B*, 53:285–339, 1991.
3. Trevor Hastie, Eyal Kishon, Malcolm Clark, and Jason Fan. A model for signature verification. Technical report, AT&T Bell Laboratories, 1992.
4. Paul R. Gerber and Klaus Müller. *Acta Cryst.*, A43:426–428, 1987.
5. Simon K. Kearsley. *J. Comp. Chem.*, 11:1187–1192, 1990.
6. A. Shapiro, J. D. Botha, A. Pastore, and A. M. Lesk. *Acta Cryst.*, A48:11–14, 1992.
7. R. Diamond. *Protein Sci.*, 1:1279–1287, 1992.
8. William R. Taylor and Christine A. Orengo. *J. Mol. Biol.*, 208:1–22, 1989.
9. William R. Taylor, Tomas P. Flores, and Christine A. Orengo. Multiple protein structure alignment. *Protein Science*, 3:1858–1870, 1994.
10. Liisa Holm and Chris Sander. *J. Mol. Biol.*, 233:123–138, 1993.
11. Andrej Šali and Tom L. Blundell. *J. Mol. Biol.*, 212:403–428, 1990.
12. D. Sankoff and J. B. Kruskal. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, 1983.
13. Osamu Gotoh. *J. Mol. Biol.*, 162:705–708, 1982.
14. Donald Bashford, Cyrus Chothia, and Arthur M. Lesk. *J. Mol. Biol.*, 196:199–216, 1987.
15. R. Diamond. *Acta Cryst.*, A32:1–10, 1976.
16. Amit P. Singh and Douglas L. Brutlag. In *ISMB-97*, pages 284–293, 1997.