Finding the Consensus Shape for a Protein Family*

L. Paul Chew† Klara Kedem‡

October 21, 2002

Abstract

We define and prove properties of the *consensus shape for a protein family*, a protein-like structure that provides a compact summary of the significant structural information for a protein family. If all members of the protein family exhibit a geometric relationship between corresponding α-carbons then that relationship is preserved in the consensus shape. In particular, distances and angles that are consistent across family members are preserved. For the consensus shape, the spacing between successive alpha carbons is variable, with small distances in regions where the members of the protein family exhibit significant variation and large distances (up to the standard spacing of about 3.8Å) in regions where the family members agree. Despite this non-protein-like characteristic, the consensus shape preserves and highlights important structural information. We describe an iterative algorithm for computing the consensus shape and prove that the algorithm converges. As a by-product we produce a *multiple structure alignment* for the family members. We present the results of experiments in which we build consensus shapes for several known protein families.

1 Introduction

There is widespread agreement that distantly related proteins often preserve three-dimensional structure similarity, even when their sequences are very different [8]. Since protein structure is better conserved in evolution than sequence, it is important to find the structural analogs of the *sequence profile* and of *multiple sequence alignment* for a protein family. This is the goal of the work presented in this paper.

Our *consensus shape* is a three-dimensional analog of the *sequence profile* that is sometimes used for multiple alignment of protein sequences. The sequence profile of a protein family is a 21 by n table. The 21 rows correspond to the 20 amino acids plus a row for *gap*, and the n columns correspond to positions along the protein sequence. An entry in the table contains the probability that a particular residue appears in a particular position along the protein sequence. Given a library of profiles, one for each existing family, and then given the amino-acid sequence of a new protein, one can often determine the new protein's family by finding the profile that most closely matches the new protein sequence. A related concept is that of the *consensus sequence* of a family of proteins. This is a single sequence of amino acids (plus the *gap* symbol) that represents

*This work was supported by NSF ITR #ACI-0085969, NSF Challenges in CISE #EIA-9726388, NSF CISE Research Infrastructure #EIA-9972853, and NSF #CCR-9988519. K. Kedem was also supported by an Israeli Defence Ministry grant.

†Computer Science Dept, Upson Hall, Cornell University, Ithaca, NY 14853 chew@cs.cornell.edu

‡Computer Science Department, Ben-Gurion University, Beer Sheva 84105, Israel, and Cornell University, klara@cs.bgu.ac.il

an entire protein family. See, for instance, [7] for a discussion of the sequence profile and the consensus sequence along with methods for computing them. Our consensus shape is most similar to the sequence profile in that it summarizes information at a particular position along the protein backbone rather than choosing a single representative for that position.

Two problems are addressed in this paper: (1) produce a single structure (often called a *core structure*) that summarizes structural information for a protein family, and (2) produce a multiple alignment of protein structures (as opposed to a multiple alignment of protein sequences without use of structural information). There are a number of papers in the scientific literature that have attacked problem (1) [3, 4], problem (2) [5], or both [6, 10, 12, 11]. Note that there is also a very large body of literature devoted to the problem of *pairwise* structural alignment.

Gelfand, Kister, Kulikowski, and Stoyanov [3] compute multiple structural alignment by working with distance matrices (a matrix of pairwise distances between the α-carbons) taking advantage of the fact that distances *within* a protein remain unchanged as the protein is translated or rotated. The method is based on finding corresponding pairs of α-carbons whose distances are preserved across the protein family. Secondary structure is used to identify potential corresponding pairs of α-carbons.

Gerstein and Altman [4] use a given multiple alignment (based on sequence) to iteratively derive a structural core by examining the variance for each group of aligned atoms.

Like us, Gerstein and Levitt [5, 6] use iterative Dynamic Programming. They use it primarily to derive a multiple structural alignment rather than a single summarizing structure. Their methods are analogs of those used for multiple sequence-based alignment. In [6], a core-finding procedure (similar to that of [4]) is used to find a core structure for a protein family.

Leibowitz, Fligelman, Nussinov, and Wolfson [10] use a technique called Geometric Hashing both to determine a multiple structural alignment and to detect a core of C_α atoms. This technique does not depend on the sequential order of the α-carbons; thus, it's suitable even for 3D structures that do not possess an inherent order. Note that our consensus shape is dependent on the sequence of the α-carbons; this dependence on sequence allows us to use Dynamic Programming and to prove the Preserved Structure Property (see Section 6).

Orengo and Taylor [12, 11] compute pairwise structural similarity by using the structural environment of each residue and by employing two levels of Dynamic Programming. Multiple alignment is achieved by aligning successive members according to their pairwise similarity scores. After each member is aligned, the multiple alignment is used to build a *consensus 3D template* that is then used for aligning the next member. The resulting template can then be used to, for instance, search for distant structural relatives in a database of protein structures.

We define the *consensus shape*, a pseudo-protein which efficiently summarizes the structural data for an entire protein family. We prove properties of the consensus shape and present an iterative algorithm to compute it. In particular, we show that the shape computed by our algorithm improves at each iteration and converges to a locally optimal solution. The algorithm uses a rough equivalent of multiple sequence alignment, but it is entirely based on shape rather than sequence.

We refer to the consensus shape as a *pseudo-protein* because it fails to have certain characteristics of real proteins. In particular, for the consensus shape, the spacing between successive α-carbons is variable, with small distances in regions where the members of the protein family exhibit significant variation and large distances in regions where the family members agree. Despite this non-protein-like characteristic, the consensus shape does preserve structural information. If all members of a protein family exhibit a geometric relationship between certain residues then that relationship is preserved in the consensus shape (see the Preserved Structure Property in Section 6). In particular, distances and angles that are consistent across family members are preserved. Thus, the consensus

shape provides a compact summary of the significant structural information for a family.

We have computed consensus shapes for a number of protein families (see Section 7, Experimental Results). Experience has shown that the algorithm's convergence is rapid. Section 7 includes example consensus shapes for several protein families as well as an example showing a consensus shape built from a group of unrelated proteins.

The following section provides terms and definitions. Section 3 provides background material on the *URMS metric* that we use to compare the shapes of protein α-carbon backbones and on the table-driven form of Dynamic Programming used to find the consensus shape. In Section 4 we describe the algorithm for finding the consensus shape. We prove convergence of the algorithm in Section 5. In Section 6 we state and prove some useful properties of the consensus shape including the Preserved Structure Property. Section 7 describes the experimental results. Conclusions and future work are discussed in the remaining, final section.

2 Terms and Definitions

Protein sequence. The sequence of amino-acids of a protein. A protein is a chain of amino-acids, thus there is a global ordering on the positions (indices) of the amino-acids of the protein.

Protein structure. The coordinates of all the atoms of a protein. In protein structure comparison it is common to take one representative atom of each amino-acid (its α-carbon) and to represent the protein's structure as a three dimensional polygonal line that connects consecutive α-carbons. This is often called the protein's *backbone*. There is a 1-to-1 match between the ordering along the sequence and that of the α-carbons of the backbone.

Unit vectors. Given a protein backbone, we compute the vector between each pair of consecutive α-carbons along the backbone. Since the distance between two consecutive α-carbons is always about 3.8Å, each vector is a *unit vector* where the unit is 3.8Å. We sometimes refer to a such a unit vector as a *direction vector*. Note that our unit vectors are all within a fixed reference frame; in other words, vectors are not taken relative to preceding vectors.

Alignment. An alignment between two proteins is a 1-to-1 mapping between the positions (indices) of their amino-acids. Notice that this definition is identical for sequence alignment and structural alignment.

Orientation. Assume we want to measure the similarity between two proteins using the RMS (or URMS) distance-measure. First we need an alignment of the two proteins, namely, we need to know which α-carbon in one protein corresponds to in the other protein. Since every protein in the PDB is measured in its own coordinate frame, we now want to find the three dimensional rigid transformation that will bring the aligned atoms as close as possible to each other under our distance-measure. Typically, such a rigid transformation involves both *translation* and *orientation*. For our URMS distance, all unit vectors are considered to share a common origin, so only orientation is involved.

Consensus shape. The consensus shape for a protein family is a sequence of vectors corresponding to the backbone of a pseudo-protein. Intuitively, each consensus vector is the average of the unit vectors, one unit vector chosen from each member of the protein family, in the corresponding

position in each protein. This intuitive picture is complicated by the presence of *gaps* which must be handled carefully to ensure that the consensus shape properly reflects structural similarity among family members.

PDB. The Protein Data Bank. A data bank of protein structures [13].

SCOP. Structural Classification of Proteins. A partly automated and partly manual data bank of proteins classified according to their structure [14].

3 Background

URMS. The distance-measure we use to construct our consensus shape is based on the *unit-vector RMS* (URMS) metric that we developed for pairwise shape-comparison of protein structures [1, 9]. In this paper, we apply and extend this measure to derive a notion of consensus shape.

The URMS distance is a variant of the RMS distance: instead of comparing residue positions we compare unit vectors. The compared vectors are those between adjacent α-carbons along the protein backbone. We refer to these direction vectors as *unit* vectors because, for proteins, these vectors are all the same length (about 3.8Å). By chaining the unit vectors head-to-tail, we obtain the standard model of a protein as a sequence of α-carbons in space. Alternatively, we can place all of the unit vectors at the origin; the protein backbone is thus mapped into vectors in the unit sphere. For the URMS distance between two proteins A and B, we first compute their representations as unit vectors mapped to the unit sphere. We then determine the orientation (orientation only, no translation) that minimizes the sum of the squared distances between corresponding unit vectors. The square root of the resulting minimum sum is defined as the URMS distance between the original proteins A and B. Note that, just like the RMS distance, the URMS distance computation provides two things: (1) the distance between the two proteins and (2) the orientation that achieves that distance.

The widely-used minimum RMS distance between two sets of atoms in $\Re^3$ is useful primarily when aligning two structures that are extremely similar. It is less appropriate for comparing structures that have only a distant similarity due to its sensitivity to outliers. This is because points far from the center of mass are, in effect, weighted more heavily than points near the center of mass. Consequently, if the proteins share a common "core", it will be difficult to detect using the minimum RMS distance. This is problematic in the case of proteins, where some crucial similarities occur near the center of mass. Because the maximum squared distance between any two unit vectors is 4, no single outlier can have a large effect on the URMS. As a result, the URMS exhibits a natural resilience to the presence of outliers such as those arising due to portions of two protein structures that are dissimilar. (For a more detailed discussion see our papers [1, 9].)

Dynamic Programming. Dynamic Programming (DP) is a widely used technique for designing algorithms (see [2], for instance). The variant we use here finds the best alignment between two sequences based on a *cost table*. For sequences of length m and n, the cost table is an m by n table in which position (i, j) holds the cost of aligning the item at position i in the first sequence to the item at position j in the second sequence. This kind of DP table is often used to compare two sequences of amino acids, but here we compare two sequences of unit vectors.

The goal is to find a least-cost path from the upper left to the lower right in the DP table. A path in a DP table consists of diagonal, horizontal, and vertical segments. A diagonal segment

through the box at position (i, j) corresponds to aligning item i in the first sequence with item j in the second sequence; the cost of this diagonal is equal to the corresponding entry in the DP table. A vertical (or horizontal) segment in the path corresponds to an insertion or deletion. The cost (i.e., the *gap cost*) for this kind of path segment must be specified, but we defer this until Section 4.

The least-cost path in this kind of table can be found in $O(mn)$ time using $O(mn)$ space where m and n are the lengths of the two sequences being compared. This can be improved to use just $O(m)$ space. See, for instance, [7] for details.

4 Finding the Consensus Shape

We present an iterative algorithm to compute the consensus shape. As outlined in the box below, our algorithm consists of two stages, the Initialization stage and the Optimization stage.

Outline of Consensus Shape Algorithm

Input: a set of proteins belonging to a single protein family; each protein is represented by the the sequence of unit vectors along its α-carbon backbone

Output: a consensus shape (a pseudo-protein) that summarizes the shape information for the family; this shape is represented as a sequence of vectors (each of length $\leq$ one unit)

Initialization:

- **I1.** Arbitrarily choose a protein to be the initial consensus shape C.
- **I2.** Align each protein with C.
- **I3.** Use the alignment to orient each protein with respect to C, minimizing its URMS distance from C.

Optimization: Loop until C quits changing:

- **L1.** Use DP to align each protein against C.
- **L2.** Improve C (using each of the 4 improvement methods described below)

End Loop.

We elaborate upon each stage of the algorithm in the corresponding subsections below.

4.1 Initialization

To start our algorithm, we arbitrarily choose some member of the protein family and then orient the others with respect to this initial shape. There are many ways that this initial orientation phase can be done. For instance, we can use an existing algorithm for multiple-alignment of amino-acid sequences. Once we have a correspondence between residues, we can then apply the minimum RMS distance to orient the proteins. However, our goal is to create a technique based entirely on shape;

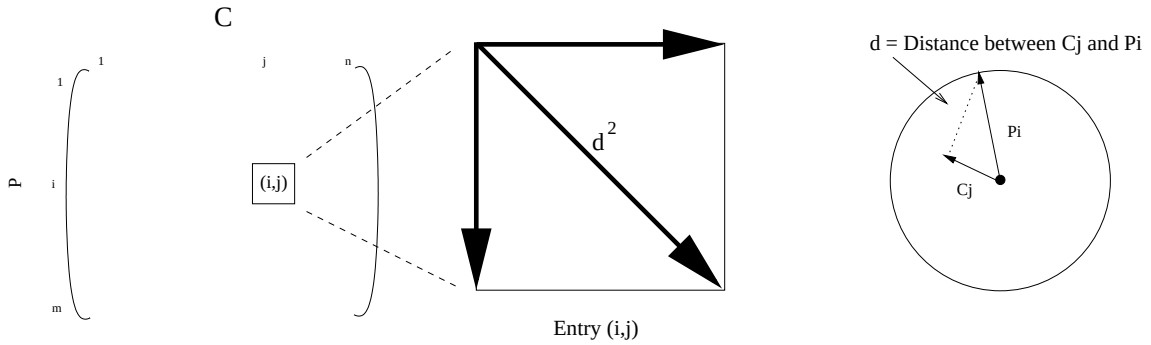

Figure 1: The DP cost table for protein P and consensus shape C at the Optimization Stage. One entry is magnified: the diagonal cost at entry (i, j) is the squared vector distance between P_i and C_j. The cost for a horizontal (a gap) is the squared distance between a gap vector and C_j. The cost for a vertical (another kind of gap) is the squared distance between a gap vector and P_i.

thus, we would prefer to avoid any initialization method that depends on the sequence of amino acids.

In our implementation, we orient each protein by comparing it with the arbitrarily-chosen initial consensus C by looking at *runs* of unit vectors where a run is some small number (any number between 5 and 8 works well) of adjacent unit vectors. Dynamic Programming is used to align these runs; an entry (i, j) in the DP matrix holds the cost for aligning a run starting at position i in in P with a run starting at position j in C. The cost of aligning this pair of runs is equal to the square of the URMS distance between them. We have chosen to set the gap cost (i.e., the cost of an unmatched run) to 0.3. The intuition is that gaps should behave like poor matches; thus, we use 0.3, indicative of a poor match when using the URMS metric. DP provides the min-cost path through our cost table; diagonals indicate places where the proteins share similar structure. These diagonals also provide a correspondence between unit vectors. So the outcome of this DP is an alignment of P with C. The aligned unit vectors of P and C are used as the basis for a URMS computation, resulting in the desired initial orientation for each protein.

4.2 Optimization: Dynamic Programming

The cost table used for Dynamic Programming in the Optimization phase uses different cost values than the cost table used in the Initialization phase. For the Optimization phase, the cost for a diagonal is simply the squared distance between the endpoints of the corresponding vectors (i.e., the squared vector difference; see Figure 1). If the corresponding direction vectors point in the same direction and are of the same length then the cost is zero. If the vectors point in opposite directions then the cost can be as high as 4.

We also must specify the costs in the table for *gaps* (horizontal and vertical). We introduce a special gap-vector in a direction perpendicular to the $3D$ coordinate axes (i.e., we use a fourth dimension). The gap-vector is a unit vector in this special gap-direction $(0, 0, 0, 1)$. The cost of a vertical in the DP table is, just as for diagonals, the squared distance between the endpoints of the corresponding vectors; in this case, the corresponding two vectors are a protein unit-vector $(p_x, p_y, p_z, 0)$ and a gap-vector $(0, 0, 0, 1)$. Since these are perpendicular unit vectors, the cost (the squared distance between the endpoints) is 2. The cost for a horizontal is defined consistently as the squared distance between the endpoints of the corresponding vectors; in this case, the vectors are a gap-vector $(0, 0, 0, 1)$ and a consensus vector (c_x, c_y, c_z, c_w). Note though that a consensus

vector can have a component in the gap-direction; thus, the cost is not always 2 as it is for verticals.

We introduce the gap component in the fourth dimension to distinguish between gaps and disagreement in our consensus representation. As the consensus is improved, updated consensus vectors are produced by averaging the corresponding protein unit-vectors (and gap-vectors) that appear in the DP tables. Without the use of the special gap-direction, it is not possible to distinguish between a consensus vector produced by averaging lots of gaps and a consensus vector produced by averaging protein unit-vectors that disagree.

Intuitively, a good path consists of many diagonals (corresponding to agreement) and a few horizontals and verticals (corresponding to gaps). A path that minimizes the cost determines an alignment between the protein and the consensus. This minimum cost is closely related to the URMS distance, since it sums all the squared pairwise distances between the aligned direction vectors

4.3 Optimization: Improving the Consensus

The consensus is improved by iteratively applying four different improvement methods; the order in which these methods are applied is not important. Assuming we have p proteins, there are p DP tables, each used to derive a single DP path representing the best alignment between one of the p proteins and the current consensus shape. These p paths are used to improve the consensus.

a) Average over a column. Each DP path can be viewed as a map taking each consensus vector (i.e., each column) to a protein unit-vector, based on the shape of the path in that column. If the DP path within a column is a diagonal then the correspondence is the obvious one and the appropriate unit vector from the protein is used. If the DP path within a column is a horizontal then the special gap-vector is used. Thus, for each column, there is one unit vector from each DP table; these unit vectors are averaged to get a new consensus vector for that column. Note that the average here includes a component in the gap-direction; thus, each consensus vector is a 4-vector.

b) Improve orientations. Each diagonal on a DP path represents a correspondence between a protein unit-vector and a consensus vector. Based on this correspondence, we run an RMS calculation on the vector endpoints to improve the protein's orientation. By definition, the RMS distance produces a new protein orientation that minimizes the sum of the squared distances between corresponding endpoints. Note that this RMS calculation is similar to our earlier URMS calculation in the sense that there is no translation involved — the vectors already share a common origin. Note also that horizontals and verticals don't affect this calculation. Horizontals have no effect because the cost of a horizontal is unchanged by rotation of the protein. Verticals have no effect because verticals are always the same cost regardless of protein orientation. Also, the components of the consensus vectors in the special gap-direction have no affect on the calculation. The sum of squared distances will have a portion due to the gap-direction components of the consensus vectors, but this portion is unchanged as proteins are rotated. In effect, the gap-components of the consensus vectors can be ignored at this stage.

c) Introduce a new column. Ideally, we want to discover an appropriate corresponding consensus vector for each unit-vector of each protein. To do this, we must eliminate all the vertical segments from the DP paths. Without verticals, we also have a Preserved Structure Property (see Section 6). If a path has a vertical segment or a contiguous run of vertical segments then we intro-

duce a new column in the consensus at that vertical's position. The initial value of the consensus vector for this column is a unit gap-vector.

d) Delete an existing column. If there is a column that, for each DP path, contains a horizontal then that column contributes no information and is thus eliminated. The consensus vector corresponding to this column is deleted.

5 Convergence

We claim that each of the four consensus-improving methods described above reduces the total cost summed over all the DP paths. Because the total cost always decreases while remaining positive, the algorithm must converge.

a) Average over a column. A new consensus vector is determined by averaging the corresponding vectors in that column of the DP tables. By definition, the new endpoint of the consensus vector is the center-of-mass of all of the endpoints of these corresponding vectors. It is easy to see, by use of some simple calculus, that the center of mass is the unique position u that minimizes the sum of the squared distances between u and each of the data points. Consider what happens if we average over a single column (assume for the moment that we continue to use the same DP paths). In this case, the total cost of all DP paths has decreased (or stayed constant) because the portion due to this column has decreased (or stayed constant) since the new consensus vector is the one that minimizes the cost for that column. Of course, we actually compute new DP paths after column averaging, but these paths can be no worse than the earlier paths and should actually be better.

b) Improve orientations. In effect, each protein is rotated to achieve a better DP path for that protein. When we sum over all paths this total sum must decrease.

c) Introduce a new column. We introduce a new column only if we have an existing vertical in some DP path. Once the column is introduced, consider the path that is exactly the same as the old DP path except for a diagonal in this column, a diagonal that is in the same row as the earlier vertical. The cost of this diagonal is 2, exactly the same as the cost for the original vertical; thus, the cost for this path is exactly the same as the cost for the earlier path. Now consider what happens to a DP path for a different protein, one that did not have a vertical corresponding to the new column. Consider a path for this protein that is exactly the same as its old DP path except for a horizontal link connecting the two portions of the earlier path across the new column. Using our rules, the consensus vector for this new column is a gap-vector; thus, the cost of the new horizontal is zero (the distance between 2 identical gap vectors). In other words, the new path with the horizontal across the new column has exactly the same cost as the old path. We have found paths in the modified DP tables that cost exactly the same as the old paths. Any new paths found by using DP must cost the same as or less than these paths; implying that the total sum over all DP paths decreases (or remains unchanged) after introducing the new column.

d) Delete an existing column. If all the DP paths for a particular column have a horizontal in that column then, when we average over the column, the consensus vector is a gap-vector. If the consensus vector is a gap vector then the cost for this column over all DP paths is zero. This column can be eliminated without changing the total cost of the DP paths.

6 Consensus Properties

Once the consensus algorithm halts we have a set of DP tables, one for each protein in the protein family. Each resulting DP table contains a DP path that consists entirely of diagonals and horizontals. Further, each protein in the family is oriented so that the cost of its DP path is minimized. Each consensus vector corresponds to a column, the same column in each DP table.

The DP paths provide some additional useful mappings. Each DP path determines a function from a consensus residue to a protein residue. Let $r(i,j)$ be the residue in protein i that corresponds to residue j in the consensus. The function r maps each consensus residue to a single residue in each protein; this map is used in the proof below. Note that this function (as a function of j for fixed i) is onto, but not 1-to-1; several consensus residues can all map to the same protein residue. Each DP path also determines a function from a protein unit-vector to a consensus vector. Let $V(i,j)$ be the consensus vector that corresponds to unit-vector j in protein i. Note that this function (as a function of j for fixed i) is 1-to-1, but not onto (i.e., for a single DP path there are some consensus vectors that are not the image of any protein unit-vector). This function maps each protein unit-vector to a single consensus vector; this map is used to determine colors used in the figures in Section 7.

We claim that if all members of a protein family exhibit a geometric relationship between certain residues then that relationship is preserved in the consensus shape. In particular, distances and angles that are consistent across family members are preserved. Thus, the consensus shape provides a compact summary of the significant structural information for a family. This Preserved Structure Property is stated more precisely in the following theorem.

Theorem 1 *(*Preserved Structure Property*): Let a and b be residue positions in the consensus and let $P(a,b)$ represent the (3D) vector from residue a to residue b. Then*

$$P(a,b) = \frac{1}{n}\sum_{i=1}^{n} P(r(i,a), r(i,b))$$

where n is the size of the protein family and r is the function (described above) mapping consensus residue to protein residue.

In other words, the vector relationship between two consensus residues is just the vector average of the corresponding relationships in all the proteins. Thus, if the members of a protein family exhibit a consistent relationship between corresponding residues then that relationship is preserved in the consensus. Note that this property holds even for residues that are widely separated along the protein backbone. If two distant residues appear with a consistent geometric relationship among all family members then that relationship is preserved in the consensus shape. Thus, all geometric similariy is preserved, not just local similarity such as that shown by, for instance, α-helices. The proof is relatively straightforward index manipulation. Note that the use of a fourth dimension for gap vectors is essential; otherwise the gap vectors interfere with the preserved structure.

Proof: Let $C(j)$ be the (3D) consensus vector from consensus residue $j-1$ to residue j. Note that a consensus vector typically has a nonzero component in the gap-direction, but this fourth coordinate is dropped here since, for this proof, we care only about 3D relationships. $P(a,b)$ is just a sum of consensus vectors:

$$P(a,b) = \sum_{j=a+1}^{b} C(j).$$

Experiment	0	1	2	3	4	5	6	7
4 globins	165	103	99	98				
16 globins	1049	558	495	486	485	483	482.37	482.24
6 alpha-beta	192	60	49					
4 tim barrels	1220	727	662	646	641	640	639.99	
6 beta	307	203	183	180				
5 mixed	988	549	523	522				

Table 1: Sums of DP costs from the initial iteration (leftmost) to the final iteration.

(a) Experiment	(b) Prot Length	(c) Iters	(d) Cons length	(e) % > 0.8
4 globins	152	3	156	66
16 globins	157	8	182	70
6 alpha-beta	170	2	170	94
4 tim barrels	435	5	509	56
6 beta	114	3	126	60
5 mixed	178	3	207	9

Table 2: A summary of experimental results. Column (a) shows the number of proteins and the name of the experiment. Column (b) shows the length of the longest protein in the experiment in unit vectors (i.e, one less than the number of residues). Column (c) shows the number of iterations until convergence. Column (d) shows the length of the final consensus shape (i.e., number of consensus vectors). Column (e) shows the percent of consensus vectors whose length is greater than 0.8 (i.e., the percentage of consensus vectors that indicate strong agreement among the proteins).

From the Consensus Algorithm, we get

$$C(j) = \frac{1}{n}\sum_{i=1}^{n} P(r(i,j-1), r(i,j)).$$

This holds because a consensus vector is just the vector average of the corresponding protein unit-vectors and corresponding gap-vectors. In this case, the gap-vectors contribute nothing since we are ignoring the gap component of the consensus vector. Note that a horizontal in column j of DP table i causes $r(i,j-1)$ and $r(i,j)$ to be the same protein residue; thus, for a horizontal $P(r(i,j-1), r(i,j))$ is a zero vector.

Combining the above equalities and reordering the sums leads to

$$P(a,b) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=a+1}^{b} P(r(i,j-1), r(i,j)).$$

But $\sum_{j=a+1}^{b} P(r(i,j-1), r(i,j))$ is just a sum of protein unit-vectors (and zero vectors); it can be rewritten as $P(r(i,a), r(i,b))$, completing the proof. ■

7 Experimental Results

We have tried our consensus algorithm on several groups of proteins. The different experiments reported here are as follows (proteins are identified by their PDB codes [13]): 4 globins (2lhb, 1eca, 2vhbA, 5mbn), 16 globins (1hlb, 1hlm, 1babA, 1babB, 1ithA, 1mba, 2hbg, 2lhb, 3sdhA, 1ash, 1flp,

1myt, 1eca, 1lh2, 2vhbA, 5mbn), 6 alpha-beta (1aa9, 1gnp, 6q21, 1ctg, 1qra, 5p21), 4 tim barrels (6xia, 2mnr, 1chr, 4enl), 6 beta (1cd8, 1ci5, 1qa9, 1cdb, 1neu, 1qfo), and 5 mixed (1cnpB, 1jhgA, 1hnf, 1aa9, 1eca). The SCOP site [14] indicates that the 4 globins and the 16 globins are from the globins family; the 6 alpha-beta are from class alpha-beta proteins, family G proteins; the 4 tim-barrels are from the tim beta/alpha fold; and the 6 beta are from the immunoglobulin (V-set domains) family.

Table 1 gives an idea of how quickly the algorithm converges. This table shows total cost (summed over all the DP Tables) for several experiments. For each row of the table, any additional iterations produce results identical to the cost appearing as the rightmost entry for that row. The algorithm converges rapidly even when the proteins are unrelated (i.e., the 5 mixed row).

Figures 2, 3, and 4 summarize the results of running our consensus-shape algorithm on the sets 4 globins, 6 beta, and 5 mixed, respectively. In each figure, the structure shown on the upper left is the resulting consensus shape, colored to show the lengths of the shape's consensus vectors. Dark vectors are long; lighter colors (i.e., red to yellow) are shorter. The graph on the upper right in each figure shows the lengths of the consensus vectors as a function of position along the protein backbone. A length near one indicates that the protein unit-vectors at this position are all in agreement (and thus the corresponding vector in the consensus shape is darkly colored). The remaining four structures in each figure are four of the proteins used to build the consensus shape. These proteins have been oriented as determined by our consensus-shape algorithm (i.e., this is the orientation for each protein that brings it into closest agreement with the consensus shape). For these structures, each unit-vector is colored to match the corresponding consensus vector; this provides a clear visual display of the parts of the protein that are common to all the members of the given set. Note that, for Figures 2 and 3, the dark portions of the protein structures look much the same. Figure 4 shows what happens when unrelated protein structures are used to determine a consensus shape; note the very small number of darkly colored vectors in this figure.

Table 2 provides a nonpictorial summary of our experiments. Note that the final consensus shape for a set of proteins is always at least as long as the longest protein in the set. This is because each protein unit-vector must be aligned to some consensus vector; thus, the minimum possible length for the consensus shape is that of the longest protein in the set. Where there is disagreement among the proteins, new columns are produced, so the final consensus shape can be longer than the longest protein.

8 Conclusions and Future Work

The consensus shape (1) can be computed efficiently using an iterative algorithm and (2) provides a concise summary of the structural information for a protein family. In particular, the consensus shape has the Preserved Structure Property: if all members of a protein family exhibit a geometric relationship between corresponding alpha carbons then that relationship is preserved in the consensus shape. In other words, distances and angles that are consistent across family members are preserved.

As well as summarizing structural information, the consensus shape provides a useful way to visualize the ways in which an individual protein differs from the other members of a protein family (see Figures 2 and 3). Since each unit-vector along a protein backbone corresponds to a single consensus vector, we can color each unit vector to match the length of the corresponding consensus vector. The length of a consensus vector indicates how strongly the various members of a protein family are in agreement. Thus, the coloring highlights the parts of the protein's structure that are common to all the members of a family.

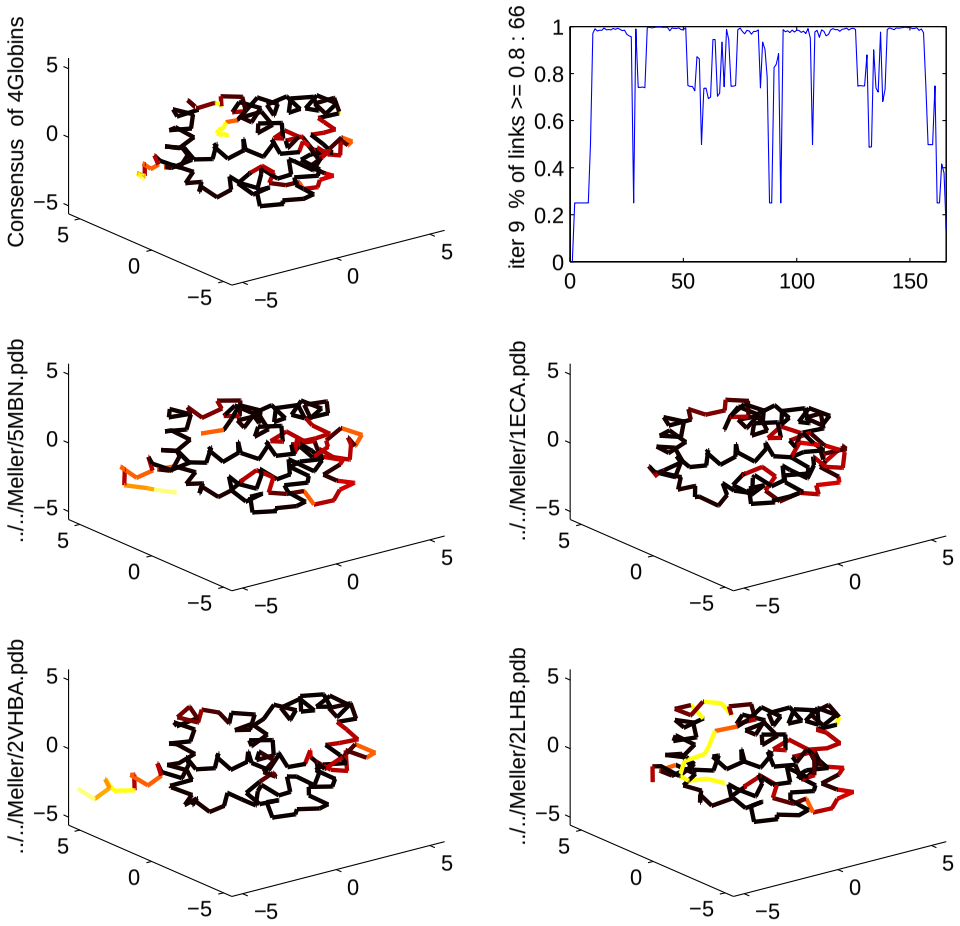

Figure 2: "4 globins" experiment. Best viewed in color.

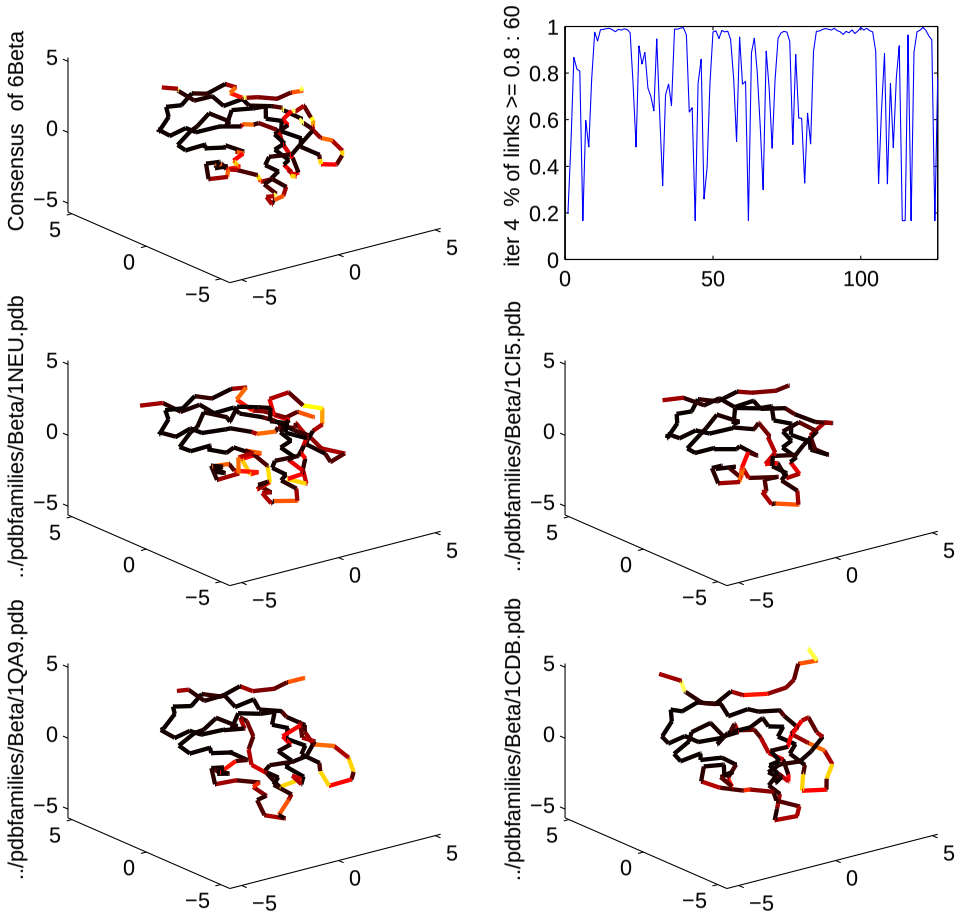

Figure 3: "6 beta" experiment. Best viewed in color.

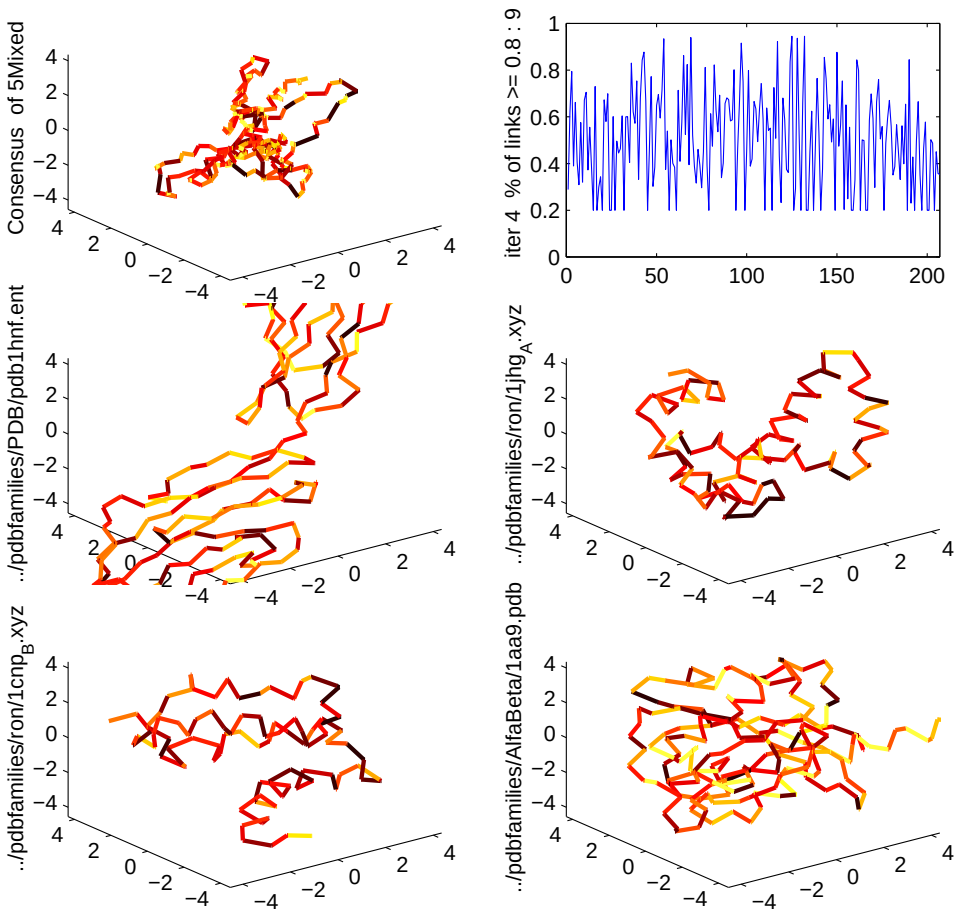

Figure 4: "5 mixed" experiment. Best viewed in color.

It is possible to construct artificial proteins that cause the Consensus Shape Algorithm to become stuck in a local rather than a global optimum. We have not detected an occurrence of this problem when using actual protein data. We tried the program using different members of a protein family as the initial consensus shape; this does not appear to affect the final resulting consensus shape.

The gap costs in the Consensus Shape Algorithm are somewhat arbitrary. Two constants are used: (1) a gap-cost of 0.3 used in the Initialization phase of the algorithm and (2) the length (currently set at 1 unit) of the fourth-dimensional gap-vector used in the Optimization phase of the algorithm. We are exploring the use of statistical tests to evaluate possible values for these constants, as well as evaluating possible values for the run-length used in the Initialization phase. Preliminary results indicate that our choices for these constants were reasonably good ones: modification of the constants appears to lead to, at best, insignificant improvements.

The Consensus Shape Algorithm, as presented here, is entirely based on shape. This reflects our own interest in geometry as well as our desire to develop an alternate, sequence-independent strategy for multiple-alignment. Because our technique is sequence independent (i.e., it does not depend on the sequence of amino acids), it is possible that a sequence-based algorithm for multiple-alignment could be used to improve our shape-based multiple-alignment. Alternately, if the proteins' structures are known, our shape-based multiple alignment could potentially improve the results of a sequence-based algorithm.

We are exploring further applications of the consensus shape. In particular, we are testing a variant of the consensus shape to determine a pairwise "distance" between proteins with the goal of automatically clustering proteins into families.

We believe that the consensus shape can improve *threading*. Threading is the process of aligning a protein sequence S (whose structure is unknown) with known protein structures in order to determine which structure is most likely for S. The consensus shape highlights the important structural features of a family and makes the threading process less sensitive to the non-essential details of individual structures.

9 Acknowledgments

The authors would like to thank Ron Elber for very helpful discussions. Thanks also to Nir Esterman and Niv Sabath for help in programming and running the tests.

References

[1] L. P. Chew, K. Kedem, D. P. Huttenlocher, and J. Kleinberg. Fast detection of geometric substructure in proteins. *Journal of Computational Biology*, 6:(3-4):313–325, 1999.

[2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill, 1990.

[3] I. Gelfand, A. Kister, C. Kulikowski, and O. Stoyanov. Geometric invariant core for the V_L and V_H domains of immunoglobulin molecules. *Protein Engineering*, 11:1015–1025, 1998.

[4] M. Gerstein and R. Altman. Average core structure and variability measures for protein families: Application to the immunoglobins. *Journal of Molecular Biology*, 112:535–542, 1995.

[5] M. Gerstein and M. Levitt. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. In *Proceedings of ISMB'96 Intelligent Systems for Molecular Biology*, pages 59–66. AAAI Press, 1996.

[6] M. Gerstein and M. Levitt. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classificatin of proteins. *Protein Science*, 7:445–456, 1998.

[7] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

[8] L. Holm and C. Sander. Mapping the protein universe. *Science*, 273:595, 1996.

[9] K. Kedem, L. Chew, and R. Elber. Unit-vector RMS (URMS) as a tool to analyze molecular dynamics trajectories. *Proteins: Structure, Function and Genetics*, 37:554–564, 1999.

[10] N. Leibowitz, Z. Fligelman, R. Nussinov, and H. Wolfson. Multiple structural alignment and core detection by geometric hashing. In *Proceedings of ISBM'99 Intelligent Systems for Molecular Biology*, pages 169–177. AAAI Press, 1999.

[11] C. Orengo and W. Taylor. SSAP: Sequential structure alignment program for protein structure comparison. *Methods in Enzymology*, 266:617–635, 1996.

[12] C. A. Orengo. CORA–topological fingerprints for protein structural families. *Protein Science*, 8:699–715, 1999.

[13] PDB: The Protein Data Bank. http://www.rcsb.org/pdb/.

[14] SCOP: Structural classification of proteins. http://scop.mrc-lmb.cam.ac.uk/scop/.