

# CTSS: A Robust and Efficient Method for Protein Structure Alignment Based on Local Geometrical and Biological Features

Tolga Can and Yuan-Fang Wang

Department of Computer Science, University of California at Santa Barbara, USA  
{tcan,yfwang}@cs.ucsb.edu

## Abstract

*We present a new method for conducting protein structure similarity searches, which improves on the accuracy, robustness, and efficiency of some existing techniques. Our method is grounded in the theory of differential geometry on 3D space curve matching. We generate shape signatures for proteins that are invariant, localized, robust, compact, and biologically meaningful. To improve matching accuracy, we smooth the noisy raw atomic coordinate data with spline fitting. To improve matching efficiency, we adopt a hierarchical coarse-to-fine strategy. We use an efficient hashing-based technique to screen out unlikely candidates and perform detailed pairwise alignments only for a small number of candidates that survive the screening process. Contrary to other hashing based techniques, our technique employs domain specific information (not just geometric information) in constructing the hash key, and hence, is more tuned to the domain of biology. Furthermore, the invariancy, localization, and compactness of the shape signatures allow us to utilize a well-known local sequence alignment algorithm for aligning two protein structures. One measure of the efficacy of the proposed technique is that we were able to discover new, meaningful motifs that were not reported by other structure alignment methods.*

## 1. Introduction

The number of known protein structures is increasing rapidly, as more researchers are joining the hunt for novel protein structures, more experimental apparatus are deployed, and more theoretical frameworks and software tools are developed for predicting protein structures. Protein structure comparison tools play an important role in this enterprise. In predicting a protein structure from its sequence, researchers usually form a new candidate structure. To avoid potential

exponential explosion of structures, that new structure is compared with previously known structures for verification/tuning/correction. Discovering similar folds or similar substructures thus provides restrictions on the conformational space and serves as a starting point for producing useful models [5].

Structure comparison is an NP-Hard problem [10]. There are no fast structural alignment algorithms that can guarantee optimality within any given similarity measure. Therefore, existing structure comparison methods employ heuristics. There are different approaches for extracting structural features. Some methods use only the coordinates of the  $C_\alpha$  atoms [13],[23]. They infer the global structure by examining the inter-atomic distances between residues. There are also quite a number of methods that use secondary structure elements (SSEs) to simplify the problem by finding initial alignments of SSEs [9],[14],[20],[24],[27] to guide the match of amino acids.

Some methods rely on localized features. In [19], a feature extraction method was proposed that examines the k-tuples of atoms in a spherical shell neighborhood of a residue. For retrieving similar structures, geometric hashing is used, which was first introduced in computer vision [18]. Geometric hashing is also used in [21] and [22]. However, those techniques are not localized as [19], and can be slow due to the large amount of redundant information kept. It can take as much as 18 seconds to compare two proteins [22]. Nevertheless, geometric hashing is the first approach that targets the need of indexing for fast similarity searches in a large structure database. Another advantage of geometric hashing is that it can also be used for multiple structural alignment [19]. However, one complaint about the pure geometrical methods is that since they do not make use of domain specific knowledge, they may overlook some biologically significant relationships such as secondary structure assignment or residue properties, e.g. hydrophobicity.

Another difficulty in the structural comparison problem is the choice of a measure to quantify the similarity between compared structures [8]. One of the widely used measures is the RMSD (root mean square distance) measure [8]. It is a measure of similarity based on the closeness of corresponding  $C_\alpha$  atoms of two protein structures. However, a match may involve only a subset of all  $C_\alpha$  atoms. I.e., there may exist biologically significant local alignments (even when the molecules do not share a global structural similarity). So the length of the alignment becomes an important measure. The information on the gaps in the alignment also gives hints on the quality of the alignment [8]. Some methods also compute the *p-value*, *e-value*, or *z-value* to quantify the statistical significance of the match [9],[13],[23].

The many variations of protein structure comparison algorithms briefly surveyed above show us that the problem of structure comparison is indeed hard. Furthermore, most of the algorithms are for pairwise comparison. I.e., they need to perform an exhaustive sequential scan of a structure database to find similar structures to a target query protein. This approach may not be feasible as the structure databases, such as the PDB [4], grow in size. Thus, fast and accurate methods for conducting structure similarity searches are needed (there are some efforts in designing methods that utilize indexing to make similarity searches more efficient, e.g., indexing DALI's distance matrices [3]).

In this paper, we present a new method for protein structure similarity search and alignment. *The main contribution is to improve on the accuracy, robustness, and efficiency of some existing techniques. The result is that our method is able to find, efficiently, meaningful structural similarities in proteins that were overlooked by other existing techniques.* Salient features of our method are that we construct signatures for structural matching that are *invariant* (i.e., they are not affected by the translation and rotation of a protein structure in space), *localized* (i.e., the signature at each residue location is completely determined by the local structure around that particular residue), *robust* (i.e., small perturbations of atomic coordinates induce small changes in the associated signatures), *compact* (i.e., the size of the signatures is  $O(n)$ ,  $n$ : number of residues), and *biologically meaningful* (i.e., we incorporate secondary structure assignment into the signature). These signatures are constructed and indexed off-line to improve query efficiency. The on-line matching process is carried out in a coarse-to-fine hierarchical manner to enable fast protein structure similarity search and detailed pairwise alignment that handles alignments with gaps. We have

implemented our method as an interactive tool that allows visual inspection of the alignment results and iterative discovery of possible suboptimal alignments that may have biological importance.

In the following sections we describe in detail our method. In Section 4 we present experimental results. And finally we conclude with future directions and discussions.

## 2. Methods

Our method is grounded in the theory of differential geometry on 3D space curve matching. It is well established in differential geometry [6] that the necessary and sufficient condition for structure isomorphism of two space curves is the correspondence of their curvature and torsion values, expressed as a function of the intrinsic arc length. Intrinsic arc length ( $s$ ) satisfies the property that  $|\dot{\mathbf{C}}(s)| = |d\mathbf{C}(s)/ds| = 1$ , where  $\mathbf{C}$  denotes the space curve. Such a parameterization is in general difficult to obtain in real world applications. However, for protein structure matching, the  $C_\alpha$  atoms along the backbone can be considered equally spaced because of the consistency in chemical bond formation. Hence, we can use the polygonal arc length between  $C_\alpha$  atoms as a convenient parameterization without loss of generality.

Because of the limited resolution of the apparatus used and noise inherent in any measurement process, the atom positions of a protein structure are imprecisely specified. In order to have robust and reliable shape signatures, smoothing of data points is needed to cope with experimentation, and resolution related errors. Approximation splines are used to smooth data points [11],[17]. Furthermore, we use variable error estimates for smoothing different type of secondary structures. This is biologically meaningful, because certain secondary structures (like *turns*) are much more likely to have errors in them.

After smoothing the  $C_\alpha$  coordinates of a protein with a polynomial spline, we compute its shape signature. The shape signature of a protein is a list of signature triplets, one for each of its residues. A signature triplet of a residue consists of its secondary structure assignment and curvature and torsion values at its  $C_\alpha$  position. These signatures are rotation and translation invariant. In other words if two different curves produce similar curvature and torsion values, then it can be concluded that they are similar (modulo rotation and translation) [6]. Curvature and torsion at a point along the curve provide localized geometrical information. The smoothing process produces a stable signature that is robust in the presence

of measurement noise. Furthermore, our method is not purely geometrical because we incorporate biological information such as the secondary structure assignment into the shape signatures. Thus, we achieve stability and robustness in the description at the expense of added computation of curve fitting and data smoothing. However, this smoothing and fitting process is performed *off-line* with a lenient time constraint. Hence, the trade-off is reasonable and beneficial.

After extracting the signatures, we build a hash table to index the space of invariant signatures. For a query protein structure, we compute its shape signatures using the same feature extraction procedure described above. Then, in the screening phase, we retrieve candidates of similar structures by using a voting mechanism based on the similarity of the hash keys. This allows efficient pruning of unlikely matching candidates, without expensive pairwise search of all proteins in the database.

For candidate proteins surviving the pruning process, we use a well-known dynamic programming algorithm (developed for sequence alignment) to align pairwise the signatures of two proteins structures. The alignment result is a set of correspondences of structurally related residues. Those corresponding  $C_\alpha$  atoms are superimposed and an RMSD value is computed for that subset. We present the user several similarity measures to help interpret the results: the dynamic programming score of local alignment, the RMSD after superimposition, the length of the alignment, and the length of the gaps in the alignment. We also present the results of the alignment visually for further inspection.

The main steps of our method can thus be summarized as follows:

For each protein in the database (an *off-line* process):

1. Calculate a spline fitting to best approximate the positions of the  $C_\alpha$  atoms.
2. Compute, *for each residue*, curvature and torsion values at the  $C_\alpha$  positions along the spline. The secondary structure assignment of that residue is also recorded in the signature.
3. Compute a hash key based on the signature and store that in a hash table.

For a query protein (an *on-line* process):

1. Repeat steps 1 to 2 above and use the shape signature to screen the candidates from the hash table. Perform the following steps only for the candidates surviving the screening process.
2. For two proteins (a candidate database protein and the query), construct the normalized scoring matrix based on the distances between extracted features.

3. Run Smith-Waterman [25] local sequence alignment algorithm on the scoring matrix.
4. Superimpose the corresponding residues using a fast least-squares solution [2],[26].
5. Report results in an interactive visual form.

For each query, the first step in the on-line process is an efficient hashing based screening of the database of proteins, and the last four steps are for comparing two protein structures pairwise (the query and a candidate). A normalized scoring matrix is created on which the dynamic programming algorithm is run. A number of local regions with the highest alignment scores are chosen as candidates of structural similarity and passed to the final step of superimposition. For those highly similar regions, we superimpose  $C_\alpha$  coordinates of the associated residues and check the RMSD of the alignment. We assign scores to them according to their lengths and RMSD values and return the best scoring alignment as the best structural alignment. In the following sections we explain each step of our method in detail.

## 2.1. Spline Approximation and Error Handling

The protein structure data are retrieved from the Protein Data Bank [4]. For each residue of the protein we obtain the 3D coordinates of its  $C_\alpha$  atoms from the PDB file. As a result, each protein is represented by approximately equi-distant sampling points in 3D space. To construct a smoothing spline best approximating those points, we use the *Java AppLib* package (<http://www.sccc.ru/matso/rozhenko/applib/>), which is an approximation library for Java.

We use the quintic spline approximation, which is for 1-D curves. For a space curve, we use 3 independent smoothing splines parameterized with respect to the polygonal arc length  $t$ . The library package constructs the quintic smoothing spline,  $\mathbf{C}(t)$ , to given data,  $\sigma(t_i)$ ,  $i = 0, \dots, n-1$ , where  $n$  is the size of the  $C_\alpha$  backbone, providing as small second derivative as possible (i.e., minimizing curvature). The method also ensures that the constructed spline does not deviate from the input data more than a given threshold by satisfying the following equation:

$$\sum_{i=0}^{n-1} w_i^{-1} \|\mathbf{C}(t_i) - \sigma(t_i)\|^2 \leq \epsilon^2 \quad (1)$$

where  $w_i$  are positive weights, and  $\epsilon$  is the maximum allowed deviation level. The larger a weight used for a

residue, the greater the deviation is allowed. For our experiments we have used  $w=0.2$  for *helices*,  $w=0.4$  for *strands*, and  $w=2.0$  for *turns*. Also notice that,  $\varepsilon$  is a measure for the *total* deviation of the spline curve from the data points. We have used a more intuitive measure,  $\varepsilon_0$ . It is an average error measure, not dependent on the length of the protein. We compute the  $\varepsilon$  in (1) with the following equation:

$$\varepsilon = \sqrt{\varepsilon_0^2 \cdot n} \quad (2)$$

Figure 1 shows an example of approximating 3D quintic spline for a small protein (1EI0:A), where individual spheres represent the  $C_\alpha$  atoms and the dark colored 3D curve passing through them is the constructed smoothing spline. The average error estimate,  $\varepsilon_0$ , is  $0.6 \text{ \AA}$ . The curvature along the spline is minimized. In addition, as seen in the figure, we allow more smoothing of the data where there is a *turn* (the top part connecting two helices) and less smoothing where there is a *helix*.

## 2.2. Feature Extraction

Curvature is defined as [6]:

$$\kappa = \left| \ddot{\mathbf{C}} \right| \quad (3)$$

And torsion is defined as:

$$\tau = \frac{1}{\kappa^2} \left[ \dot{\mathbf{C}} \quad \ddot{\mathbf{C}} \quad \dddot{\mathbf{C}} \right] \quad (4)$$

where the square brackets have the special meaning of:

$$\left[ \dot{\mathbf{C}} \quad \ddot{\mathbf{C}} \quad \dddot{\mathbf{C}} \right] = \begin{vmatrix} \dot{C}_x & \dot{C}_y & \dot{C}_z \\ \ddot{C}_x & \ddot{C}_y & \ddot{C}_z \\ \dddot{C}_x & \dddot{C}_y & \dddot{C}_z \end{vmatrix} \quad (5)$$

In other words, the curvature of a point on the curve denotes how rapidly the curve pulls away from the tangent at that point, or how non-colinear a curve is. Similarly, the torsion of a point on the curve denotes how rapidly the curve pulls away from the osculating plane at that point [6], or how non-planar a curve is. We

compute the average curvature and torsion in a close neighborhood of a residue. Since computation of torsion involves the third derivative of the spline polynomial, we have used quintic spline approximation, which guarantees the fourth order derivative continuity.

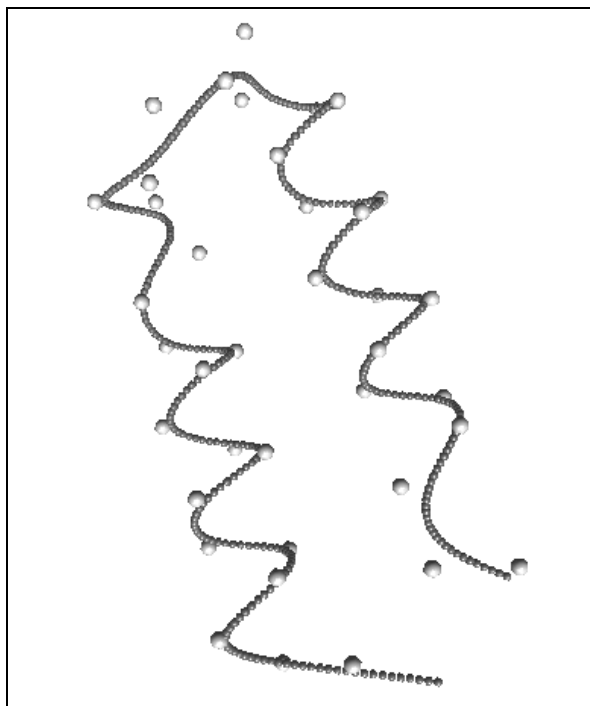


Figure 1. Spline approximation for  $C_\alpha$  coordinates

We also use the secondary structure assignment of a residue as a structural feature. Secondary structure assignment information is retrieved from the PDB web site [4]. PDB uses the DSSP method [16] to determine the secondary structure assignments of proteins. The signature value regarding the secondary structure assignment is one of the *helix*, *turn*, or *strand*. It should be noted that other biological properties of a residue, such as hydrophobicity, can also be used as part of the signature – if discrimination based on those traits are desirable.

## 2.3. Hashing for Fast Retrieval of Candidates

After the feature extraction phase, we perform a quantization and normalization procedure on the curvature and torsion values. After that procedure each curvature and torsion value resides in the interval  $[0,255]$ .

Each signature feature represents one dimension in our hash table. Therefore, we create a three dimensional

hash table for curvature, torsion, and secondary structure type. To ensure a robust and reliable retrieval of candidates, the resolution of the hash table must be judiciously chosen. The coarser the resolution, the smaller the size of the hash table gets. However, coarser resolutions reduce the discriminating power of the curvature-torsion descriptor and result in more false positive candidates surviving the screening process. On the other hand, finer resolutions increase the size of the hash table. The descriptor also becomes more susceptible to random error fluctuation, and result in true positives being screened out. After some experimentation, we have chosen the resolution of our hash table to be  $64 \times 64 \times 3$ .

For each signature triplet,  $(\kappa, \tau, ss)$ , we compute a hash key, which is simply  $(\kappa/4, \tau/4, ss)$ . By using that key as an index to the hash table, we store the signature triplet into the hash table along with its host protein chain identifier and its residue number. This process is executed *offline* for each protein in the database.

For a query protein,  $p$ , we extract its shape signatures as described in sections 2.1 and 2.2. We perform similar quantization and normalization of the curvature and torsion values. For each residue of the query protein, we compute a hash key using its signature triplet,  $(\kappa_p/4, \tau_p/4, ss_p)$ . We retrieve the hash table entry indexed by that key. We accumulate a vote for each database protein stored in that entry. We repeat this process for each residue of the query protein. At the end, we normalize the accumulated votes of database proteins by their length (number of residues). We screen out the proteins whose normalized vote is below a certain threshold,  $Z$ . In our experiments, we have chosen the threshold value  $Z$  to be  $2.0$ . In general, the threshold value should be chosen based upon the size of the smallest features that one would like to align. For this value of  $Z$ , for a query protein, on the average, 80% of the entire database is screened out after this process.

By using this voting mechanism, we efficiently retrieve candidates of similar structures. Hence, we avoid exhaustive scan of the entire database.

## 2.4. Pairwise Comparison

We perform pairwise comparison only for the candidates surviving the screening process. For two proteins under consideration, we first construct a normalized scoring matrix and then use a modified version of Smith-Waterman [25] dynamic programming algorithm on that distance matrix. The best scoring local alignment defines a set of correspondences of residues,

which is then superimposed by a fast closed-form solution. The details of pairwise comparison are explained below.

**2.4.1. Distance Matrices.** The distance matrices we compute should not be confused with the inter-atomic distance matrices of the DALI method [13]. The distance matrices we compute are the signature distance matrices between two proteins. The entry  $d_{ij}^{AB}$  of the distance matrix denotes the distance between the quantized and normalized signature values of the  $i^{\text{th}}$  residue of protein  $A$  and the  $j^{\text{th}}$  residue of protein  $B$ , and it is defined by the following equation:

$$d_{ij}^{AB} = \sqrt{(\kappa_i^A - \kappa_j^B)^2 + (\tau_i^A - \tau_j^B)^2} + s_{ij}^{AB} \quad (6)$$

$$s_{ij}^{AB} = \begin{cases} c & \text{if } SSA(r_i^A) \neq SSA(r_j^B) \\ -c & \text{if } SSA(r_i^A) = SSA(r_j^B) \end{cases} \quad (7)$$

where  $SSA(r_i^A)$  denotes the secondary structure assignment of the  $i^{\text{th}}$  residue of protein  $A$ . This measure is basically the Euclidian distance between the curvature and torsion tuples,  $(\kappa_i^A, \tau_i^A)$  and  $(\kappa_j^B, \tau_j^B)$ , regulated by the secondary structure assignment agreement. Agreement on secondary structure assignment decreases the distance by a constant  $c$ , and disagreement increases the distance by the same constant. We have used  $c=20$  in our experiments. Our choice of incorporating  $s_{ij}^{AB}$  into the signature as an offset instead of a multiplication factor is because the secondary structure assignments are not 100% correct, and may mislead the alignment if they are the dominant factor in the distance equation. Figure 2 shows an example distance matrix for the shape signature relationships between the proteins 1FAZ:A and 1YTF:D. Darker regions indicate higher similarity of signatures.

We convert those distance values to normalized score values in the interval  $[low, high]$  to be used by the local alignment in the dynamic programming phase. This is done by using the following equation:

$$score_{ij}^{AB} = low + \frac{-d_{ij}^{AB} + (256\sqrt{2} + c)}{256\sqrt{2} + 2c} \cdot (high - low) \quad (8)$$

We have chosen the *low* score to be  $-10.0$  and the *high* score to be  $20.0$ , because they define a range similar to that of the PAM matrix [7]. The score of the

alignment makes sense this way by comparing it to the sequence alignment scores.

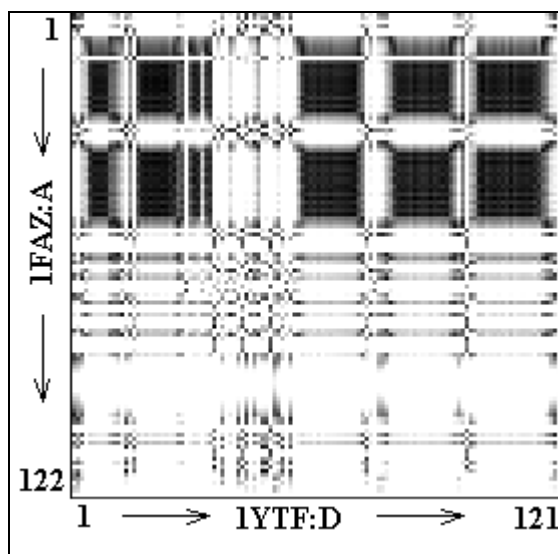


Figure 2. The distance matrix

#### 2.4.2. Local Alignment by Dynamic Programming.

The shape signatures can be thought of as protein sequence data with the alphabet,  $\Sigma = \{\text{all possible triplets of Curvature, Torsion, and Secondary Structure assignment}\}$ . We define a similarity score between two signature values, eq. (8), which is analogous to the scoring matrices such as PAM [7] and BLOSUM [12] that define similarity scores between residue types. However, we do not compute a static scoring matrix, instead a distance matrix for each pair of proteins is created as explained in Section 2.4.1.

We then run the dynamic programming algorithm for sequence alignment by Smith and Waterman [25] using the dynamically computed and normalized scoring matrix. As in local sequence alignment we use an affine gap cost model, in which opening and extending gaps have different costs. For our experiments, we have used an opening gap penalty of 14 and an extending gap penalty of 10.

The complexity of alignment by using this method is  $O(mn)$ , where  $m$  and  $n$  are the numbers of residues in the compared proteins respectively. Figure 3 shows the best local alignment of 1FAZ:A and 1YTF:D on the distance matrix and the detection of a HTH motif shared between those structures (3D result seen in Figure 4).

However, the best local alignment returned by the algorithm is not guaranteed to be the best structural alignment. Because of gaps in the alignment, the sub-structures represented by the alignment may actually

have a high RMSD (e.g., those gaps may be regions of twists and turns affecting the overall alignment). Thus, we superimpose the query protein on the database protein and check the RMSD values of a number of best local alignments to obtain the best local alignment.

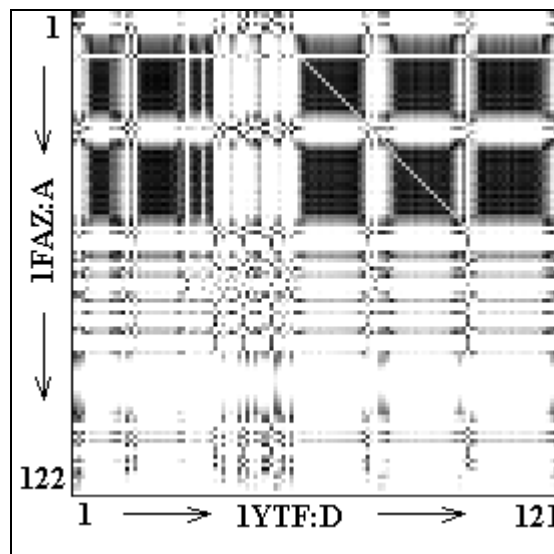


Figure 3. Local alignment with best score

#### 2.4.3. Superimposition.

We use a fast least-squares solution to superimpose an ordered corresponding set of points in 3D space. Given a minimum of three pairs of point correspondences, the best rotation and translation,  $\mathbf{R}$  and  $\mathbf{T}$ , can be computed efficiently in  $O(n)$  time, where  $n$  is the number of corresponding points. A non-iterative least-squares solution based on the singular value decomposition (SVD) was suggested by Arun et al. [2] to find a closed-form solution. Umeyama [26] provided modifications to [2] to ensure that a correct rotation matrix, instead of a reflection, is computed when the data are noisy.

With the help of the superimposition, we compute the minimum RMSD values of the top local alignments. We assign a score to each alignment by using the following equation:

$$SCORE = \frac{\text{length of alignment}}{RMSD \text{ of alignment}} \quad (9)$$

We return the best scoring alignment as the best structural alignment between the compared protein structures. Figure 4 shows the superimposed result of the best local alignment found for 1FAZ:A and 1YTF:D. That alignment reveals a Helix-Turn-Helix motif shared

between those protein structures. The Helix-Turn-Helix motif is usually found in DNA-binding proteins, and consists of a recognition helix and a stabilizing helix separated by a short loop.

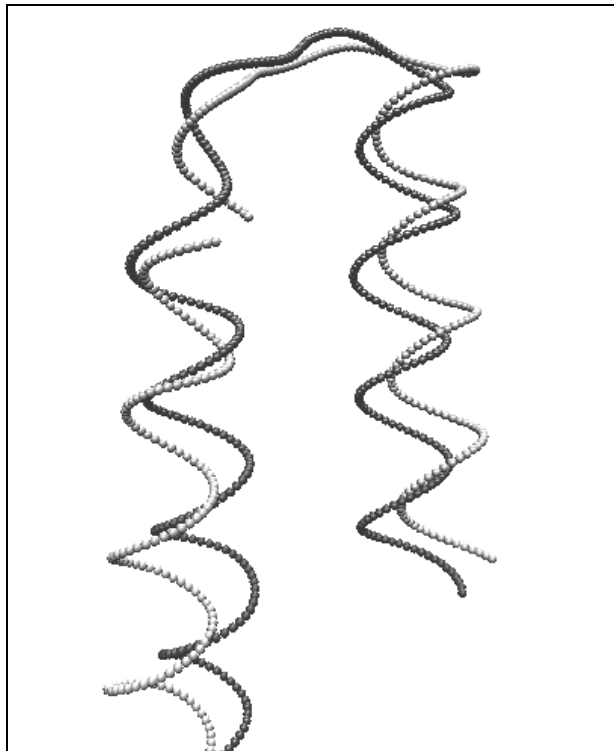


Figure 4. Superimposed local alignment result

### 3. Interactive Visualization of the Results

We use Java 3D graphics library to visualize the results (<http://java.sun.com/products/java-media/3D/>). The main advantage is that we do not have to generate visualization scripts and invoke external visualization tools like RASMOL in order to visualize the alignment results. Furthermore, our tool is platform independent and can be run within a web browser. Both the aligned 3D structures and the shape signature distance matrices are presented to the user. The result of the alignment is also shown to the user on the distance matrix. The user can select other suboptimal alignments and inspect their biological significance. Figure 5 shows the user interface with the shared motif between 2CRO:\_ and 2WRP:R.

### 4. Experiments

For our experiments, we have used a representative set of proteins selected using the PDBSELECT method

[15]. The PDBSELECT database is a subset of the structures in the PDB that does not contain homologue sequences, i.e. no two proteins have more than 25% sequence identity. Low sequence homologues present challenges to structure alignment algorithms, as it is not possible to use sequence similarity to predict structure similarity. There are 1949 protein chains in that representative database (December 2002 version)<sup>1</sup>. The total number of residues in the representative database was 314165. Because of the compactness of the signatures, the size of the hash table for the entire database is small and can fit in the memory easily. The size of the hash table for our dataset was ~3MB. The size of the hash table is considerably compact compared to the size of the representative structure database, which is approximately 300MB.

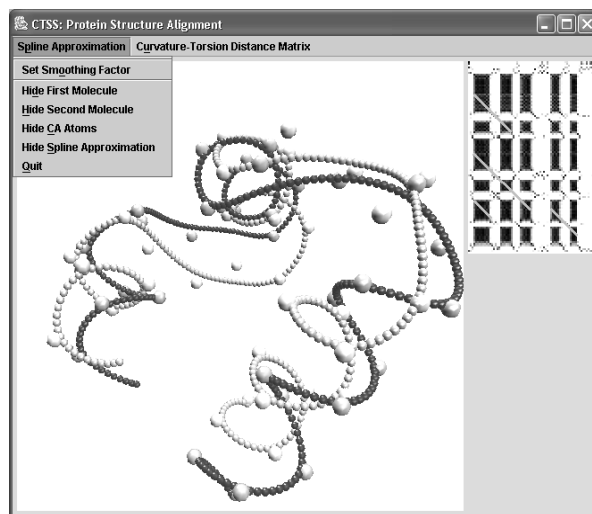


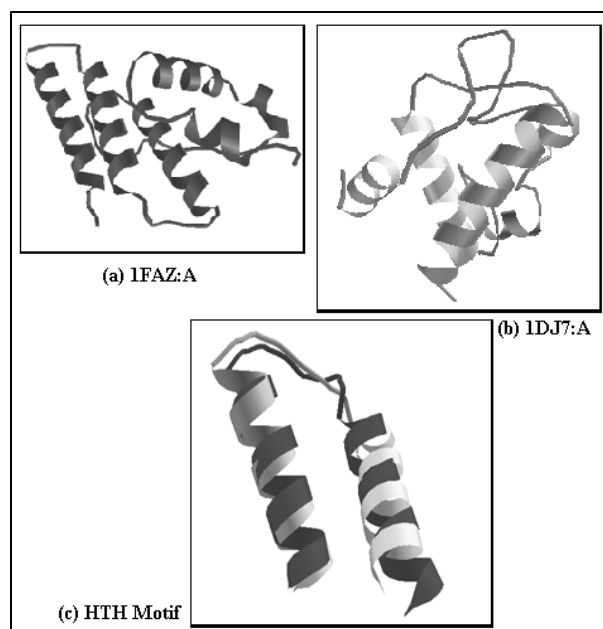
Figure 5. The user interface for CTSS

Comparing the performance of our algorithm with other structural alignment algorithms is not always possible. Most of them are provided as web services, in which the results are reported back by an e-mail notification. Time delay between submitting a query and obtaining the results back can be due to a variety of factors, which complicates timing comparison. In these web-based systems, precomputed similarity results for existing proteins can be efficiently retrieved. But structure similarity query for a new structure can take a long time, since they need to perform an exhaustive scan of the database to perform the query. To give an idea of how long it can take, as an example, DALI interactive database search may respond to a query in 5-10 minutes

<sup>1</sup> <http://homepages.fh-giessen.de/~hg12640/pdbselect/>

or 1-2 hours depending on whether the query structure has a sequence homologue in the database or not.

For our experiments, we have performed queries for the protein chains 1FAZ:A and 1B16:A. The timing results were obtained on a Microsoft Windows XP machine with Intel Pentium 4 Processor at 2.0GHz and 512MB of RAM. However, we have dedicated only 64MB of this as the maximum size of memory allocation pool for the Java Virtual Machine executing our program. For 1FAZ:A, the screening process took 18 seconds. After the screening process we selected 262 proteins structures (only 13% of the entire database) to be examined pairwise against 1FAZ:A. Those selected structures are the ones that passed the threshold ( $Z=2.0$ ) of the normalized number of votes (normalized by length). The pairwise comparison step took 29 seconds. So, the structure similarity query took a total time of 47 seconds. For 1B16:A, the screening process took 25 seconds. It took longer mainly because 1B16A is a longer protein than 1FAZ:A. After the screening process, 483 proteins (24% of the database) passed the normalized voting threshold of  $Z=2.0$ . The pairwise comparison with those 483 proteins took 68 seconds, making a total query time of 93 seconds. Below, we present selected results from those queries.



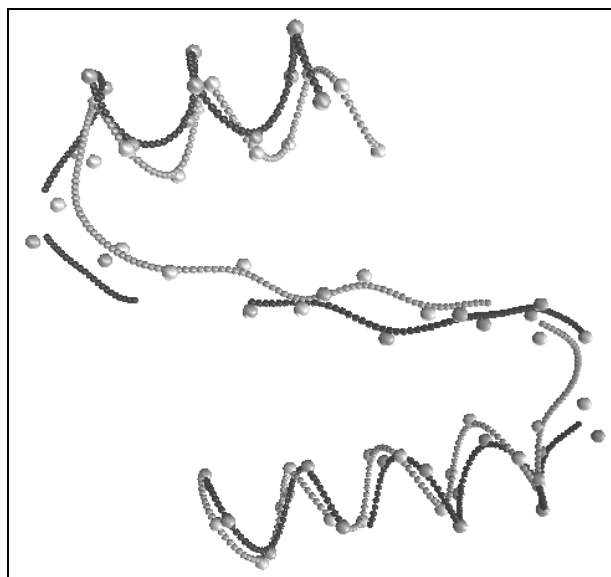
**Figure 6. Helix-Turn-Helix match between 1FAZ:A and 1DJ7:A**

For the protein chains 1FAZ:A and 1YTF:D, we have found that they share a Helix-Turn-Helix motif, with length 42, and with RMSD  $2.8 \text{ \AA}$ . Those structures

shared only 1.9% sequence identity globally. The structural alignment program CE can find this alignment with length 52, but with much higher RMSD of  $4.4 \text{ \AA}$ . The result of that alignment is depicted graphically in Figure 4.

We have also discovered some motifs not detected by other alignment tools, such as CE [23] or DALI [13]. For example, we have found the Helix-Turn-Helix motif between 1FAZ:A and 1DJ7:A, with length 38 and RMSD  $3.68 \text{ \AA}$ , and with 2 gaps in 1DJ7:A. Figure 6 shows the two proteins separately and the shared motif between them.

Another motif that was not detected by others and discovered by our program was between 1B16:A and 1H05:A. The length of that Helix-Strand-Helix motif is 35 with RMSD  $3.26 \text{ \AA}$ . Figure 7 shows that motif.



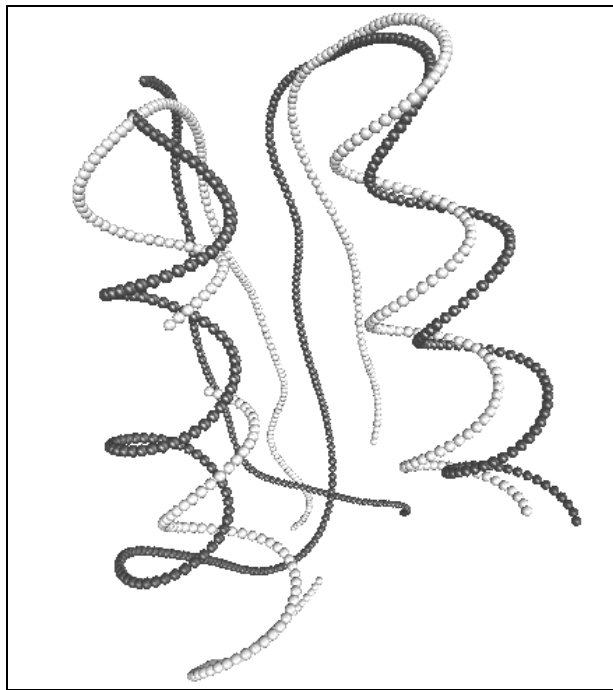
**Figure 7. Shared motif between 1B16:A and 1H05:A**

We have conducted a comparison between 1B16:A and 1GCI:\_. They share a Helix-Strand-Helix-Strand (HEHE) motif and the length of the alignment was 46, with RMSD  $3.34 \text{ \AA}$ . Those protein chains had 8.5% sequence identity. That shared motif can be seen in Figure 8.

Our program does not only find small motifs between protein structures. In our test cases we have also found longer structural alignments. 1B16:A and 1OAA:\_ alignment had length 209 with RMSD  $4.6 \text{ \AA}$ .

Figure 9 shows the Strand-Helix-Strand motif discovered between 1B16:A and 1QP8:A. We have found a substructure match of length 35, with RMSD

1.58 Å, and with two gaps of length one. Those proteins share 8.1% sequence identity.



**Figure 8. The Helix-Strand-Helix-Strand motif between 1B16:A and 1GCI:\_.**

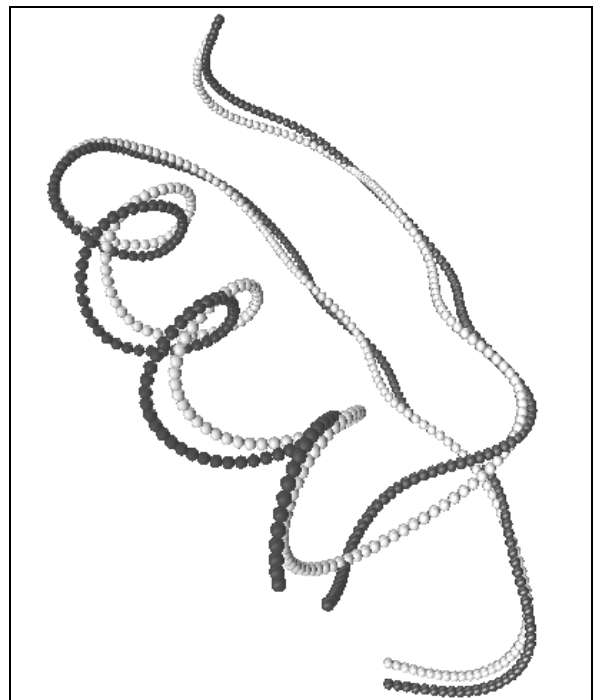
Finally, we have also conducted a pairwise comparison test between the structure 2CRO:\_ and 2WRP:R. Those structures were previously compared by Pennec and Ayache [22]. We have found a longer motif between those structures compared to what they reported and moreover our program can detect that shared substructure in just a fraction of a second (compared to 18 sec reported in [22]). The alignment result can be seen in Figure 5.

## 5. Discussions and Future Work

The novelty of our proposed technique lies in the methods of feature design, extraction, and smoothing. Together, these methods ensure the success of the ensuing phases of protein structure screening and pairwise alignment. Extracting localized features and embedding both geometric and biological information in the signature are the major difference compared to other structural alignment methods. We have also employed an efficient screening phase based on hashing followed by an accurate pairwise alignment phase that can handle gapped alignments efficiently.

However, one question that comes to mind is how descriptive these localized signatures ( $O(n)$ ) are in representing the structure of a protein? It's a fair question, especially when one considers that other existing algorithms generate signatures in the order of  $O(n^2)$  or  $O(n^3)$  to capture the structure of a protein of length  $n$ . Here, we have the support of the theory of differential geometry, which states that space curves generating the same curvature, torsion values are isomorphic. It may be argued that the existence of gaps along the alignment of curves may fail that theory. However, as the computation of curvature and torsion is a localized process, local isomorphism can still be detected based on such localized signatures. Our initial experiment results proved that our technique is a promising one. We have been able to find shared motifs between protein structures efficiently. As a future work, we are going to conduct more comprehensive tests to assess the accuracy and efficiency of our technique.

We will also investigate indexing methods other than hashing for faster screening of candidates of structural similarity. The assessment of accuracy and reliability of the indexing techniques is another critical issue that needs to be addressed.



**Figure 9. The Strand-Helix-Strand Motif between 1B16:A and 1QP8:A**

On the other hand, the analogy between sequence and structure comparison introduced by our method promises other benefits. Existing sequence comparison algorithms like BLAST [1], which can conduct very efficient sequence similarity searches, can be tailored to conduct structure similarity searches (or multiple structural alignments) with the use of our localized shape signatures that is a one to one mapping to the sequence (i.e., each residue has one localized signature value).

## 6. Conclusions

In this paper, we have presented a new method for protein structure alignment. Our method comprises a novel technique for extracting compact and localized shape signatures for protein structures, an indexing component based on hashing to avoid an exhaustive scan of the entire structure database, and a pairwise alignment method for accurately aligning shape signatures even in the presence of gaps. The shape signatures we generate are robust and stable (i.e. they are not affected by small changes in atomic coordinates) and biologically meaningful (we incorporate secondary structure assignment of a residue to the signature). Our experiments showed that our technique is able to execute protein structure similarity searches efficiently and discover biologically meaningful motifs shared between protein structures that were overlooked by other methods.

## 7. Availability

CTSS is still under development. However, the pairwise comparison component with interactive visualization of the alignment results, is freely available with source code at the following URL:

<http://www.cs.ucsb.edu/~tcan/CTSS/>

## 8. References

- [1] Altschul S.F., Gish W., Miller W., Meyers E.W., and Lipman D.J., "Basic Local Alignment Search Tool", *Journal of Molecular Biology*, **215**(3): 403-410, 1990.
- [2] Arun, K.S., Huang, T.S., and Blostein, S.D., "Least-Squares Fitting of Two 3-D Point Sets", *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, **9**(5): 698-700, 1987.
- [3] Aung, Z., Fu, W., and Tan, K.L., "An Efficient Index-based Protein Structure Database Searching Method", *In Proc. of the 8<sup>th</sup> International Conference on Database Systems for Advanced Applications (DASFAA)*, 2003.
- [4] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E., "The Protein Data Bank" *Nucleic Acids Research*, **28**(1): 235-242, 2000.
- [5] Blundell, T.L. and Johnson, M.S., "Catching a common fold", *Protein Science*, **2**(6): 877-883, 1993.
- [6] do Carmo, M.P., "Differential Geometry of curve and surfaces", *Prentice-Hall, Englewood Cliffs, New Jersey*, 1976.
- [7] Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C., "A model of evolutionary change in proteins", *Atlas of Protein Sequence and Structure*, **5**(3): 345-352, 1978.
- [8] Eidhammer, I., Jonassen, I., and Taylor, W.R., "Structure Comparison and Structure Patterns", *Journal of Computational Biology*, **7**(5): 685-716, 2000.
- [9] Gibrat, J.F., Madej, T., and Bryant, S.H., "Surprising similarities in structure comparison", *Current Opinion in Structural Biology*, **6**(3): 377-385, 1996.
- [10] Godzik, A., "The structural alignment between two proteins: Is there a unique answer?", *Protein Science*, **5**(7): 1325-1338, 1996.
- [11] Guéziec, A. and Ayache, N., "Smoothing and Matching of 3-D Space Curves", *International Journal of Computer Vision*, **12**(1): 79-104, 1994.
- [12] Henikoff, S. and Henikoff, J.G., "Amino acid substitution matrices from protein blocks", *Proc. Natl. Acad. Sci. U.S.A.*, **89**: 10915-10919, 1992.
- [13] Holm, L. and Sander, C., "Protein Structure Comparison by Alignment of Distance Matrices", *Journal of Molecular Biology*, **233**(1): 123-138, 1993.
- [14] Holm, L. and Sander, C., "3-D Lookup: Fast Protein Structure Database Searches at 90% Reliability", *In Proc. of the 3<sup>rd</sup> International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 179-187, 1995.
- [15] Hobohm, U., Scharf, M., Schneider, R., and Sander, C., "Selection of a representative protein data sets", *Protein Science*, **1**(3): 409-417, 1992.
- [16] Kabsch, W. and Sander, C., "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features", *Biopolymers*, **22**(12): 2577-637, 1983.
- [17] Kishon, E., Hastie, T., and Wolfson, H., "3-D Curve Matching Using Splines", *Journal of Robotic Systems*, **8**(6): 723-743, 1991.
- [18] Lamdan, Y. and Wolfson, H.J., "Geometric hashing: a general and efficient model-based recognition scheme", *In Proc. of the 2<sup>nd</sup> International Conference on Computer Vision (ICCV)*, 238-249, 1988.
- [19] Leibowitz, N., Fligelman, Z.Y., Nussinov, R., and Wolfson, H.J., "Multiple Structural Alignment and Core Detection by Geometric Hashing", *In Proc. of the 7<sup>th</sup> International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 169-177, 1999.
- [20] Lu, G., "TOP: a new method for protein structure comparison and similarity searches", *Journal of Applied Crystallography*, **33**(1): 176-183, 2000.

- [21] Nussinov, R. and Wolfson, H.J., "Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques", *Biophysics*, **88**: 10495-10499, 1991.
- [22] Pennec, X. and Ayache, N., "A geometric algorithm to find small but highly similar 3D substructures in proteins", *Bioinformatics*, **14(6)**: 516-522, 1998.
- [23] Shindyalov, I.N., Bourne, P.E., "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path", *Protein Engineering*, **11(9)**: 739-747, 1998.
- [24] Singh, A.P. and Brutlag, D.L., "Hierarchical Protein Structure Superposition using both Secondary Structure and Atomic Representations", *In Proc. of the 5<sup>th</sup> International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 284-293, 1997.
- [25] Smith, R. and Waterman, M.S., "Identification of common molecular subsequences", *Journal of Molecular Biology*, **147(1)**: 195-197, 1981.
- [26] Umeyama, S., "Least-squares estimation of transformation parameters between two point patterns", *IEEE Trans. Pattern Analysis and Machine Vision*, **13(4)**: 376-380, 1991.
- [27] Young, M.M., Skillman, A.G., and Kuntz, I.D., "A rapid method for exploring the protein structure universe", *Proteins: Structure, Function, and Genetics*, **34(3)**: 317-332, 1999.