IDENTIFYING STRUCTURAL MOTIFS IN PROTEINS

ROHIT SINGH[a b]
Accelrys Inc, San Diego

MITUL SAHA
Department of Mechanical Engineering, Stanford University

In biological macromolecules, structural patterns (motifs) are often repeated across different molecules. Detection of these common motifs in a new molecule can provide useful clues to the functional properties of such a molecule. We formulate the problem of identifying a given structural motif (pattern) in a target protein (example) and discuss the notion of complete matches vis-a-vis partial matches. We describe the precise error criterion that has to be minimized and also discuss different metrics for evaluating the quality of partial matches. Secondly, we present a new polynomial time algorithm for the problem of matching a given motif in a target protein. We also use the sequence and (if available) secondary structure information to annotate the different points in motif and the target protein, thus reducing the search space size. Our algorithm guarantees the detection of a perfect match, if present. Even otherwise, the algorithm computes very good matches. Unlike other methods, the error minimized by our algorithm directly translates to root mean square deviation (RMSD), the most commonly accepted metric for structure matching in biological macromolecules. The algorithm does not involve any preprocessing and is suitable for the detection of both small and large motifs in the target protein. We also present experiments exploring the quality of matches found by the algorithm. We examine its performance in matching (both full and partial) active sites in proteins.

1 Introduction

Before we can go very far in modeling and manipulating proteins *in-silico*, we need to develop quantitative measures of structural similarities between (parts of) two proteins. In the simplest case— comparing two conformations of the same protein— one usually calculates the root mean squared deviation (RMSD) between the two structures. Suppose, however, we would like to know if some part of the given protein matches a particular sub-structure. For example, finding a particular *active site* inside a protein can provide insights into the protein's function. Most of the current approaches for identification of active-sites rely on building sophisticated sequence-based models to build a 'consensus' representation of the active site[1,2]. These sequence based representations, however, are only an approximation to the underlying structural information. A method based on structure-based matching would be much more useful. The usefulness of such a technique is not restricted to small motifs ($\simeq$ 3-10 amino acids) only. Often, even larger sub-structures ($\geq$ 15 amino acids) are conserved across different proteins. These larger motifs often form *sub-domains* in a large protein. Proteins sharing such common sub-domains often share similar structural and functional properties[3,4].

The problem of finding a good match for a pattern (motif) in an example (protein) has two parts. The first part is finding the best match for the pattern in the example.

---

[a] This work was done while the author was at the Department of Computer Science, Stanford University
[b] Corresponding Author: `rohitsi@cs.stanford.edu`

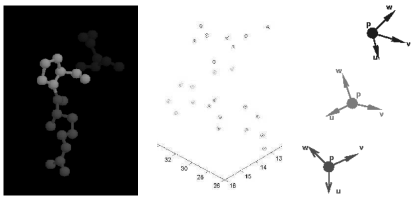
The second part is to evaluate if this match is significant enough: e.g., finding a close match for a pattern of 10 amino acids is more significant than finding a close match for a pattern for 2 amino acids. The process of evaluating the quality of a match, however, is somewhat subjective and depends on the goal of the biologist. There has only been a little work on this subject[5,6]. In the rest of this chapter, we shall restrict our focus to the first sub-problem: finding the best solution for the given input. We first formulate the problem of finding an optimal match for a pattern in a given example. We then discuss a method for solving the problem and evaluate it.

2 Problem Description

First, we introduce the notion of a *multipoint*.

Definition 1 *A multipoint* $\mathbf{a} \in \mathcal{M}$ *is an abstraction of a collection of points in* $\Re^3$ *defined as* $\mathbf{a} = \langle \vec{p}, \langle \hat{u}, \hat{v}, \hat{w} \rangle \rangle$ *where* $\vec{p}, \hat{u}, \hat{v}, \hat{w} \in \Re^3$*;* $\langle \hat{u}, \hat{v}, \hat{w} \rangle$ *define a right-handed reference frame with its origin at* $\vec{p}$ *(also referred to as the* anchor*);* $\mathcal{M}$ *is set of all multipoints and* $\Re$ *is the set of real numbers. Every multipoint* $\mathbf{a}$ *has a* $label$ $l_{\mathbf{a}} \in \mathcal{L}$ *where* $\mathcal{L}$ *is the set of all possible labels. Let* $\mathcal{T}$ *be the set of all rigid-body transformations (rotations and/or translations) in* $\Re^3$*. Applying* $T \in \mathcal{T}$ *on* $\mathbf{a}$ *is the same as applying* T *on the points which* $\mathbf{a}$ *represents:* $T(\mathbf{a}) = \mathbf{a}' = \langle T(\vec{p}), \langle T(\hat{u}), T(\hat{v}), T(\hat{w}) \rangle \rangle$*.*

Definition 2 *With a multipoint* $\mathbf{a} \in \mathcal{M}$*, a* **distance measure** *,* $D_{\mathbf{a}}(\mathbf{b})$*, can be defined as providing its distance from some other multipoint* $\mathbf{b}$ *which has the* <u>*same label*</u> *as* $\mathbf{a}$*. The calculation of this distance may depend on the internal representation of the two multipoints. The distance measure associated with* $\mathbf{a}$*, if any, should be preserved under transformations i.e.,* $D_{\mathbf{a}}(\mathbf{b}) = D_{T(\mathbf{a})}(T(\mathbf{b}))$



(a) (b) (c)

Figure 1: **Multipoints**

The above figures show how a multipoint would typically be constructed. (a) a protein structure (b) can be broken into a collection of 'significant' points. For each group of points that belong to one feature, (c) we can construct an anchor and a reference frame.

A *pattern set* $P \subset \mathcal{M}$, $|P| = m$, is a set of m multipoints. Each multipoint $\mathbf{p_i} \in P$ has an associated distance measure $D_{\mathbf{p_i}}()$. An *example set* $Q \subset \mathcal{M}$, where $|Q| = n$ and $n \geq m$, is a set of n multipoints. Given P and Q, we can define the set of possible correspondences between their multipoints:

$$\mathcal{C}_{PQ} = \{\langle r_1, r_2, \ldots, r_m \rangle | r_i \in \{1 \ldots n\};\ r_i \neq r_j \forall i,j;\ and\ label(\mathbf{p_i}) = label(\mathbf{q}_{r_i})\}$$

Notation: Henceforth, we shall use $X[i]$ to refer to the i^{th} element of the ordered set X.

Each correspondence $C \in \mathcal{C}_{PQ}$ induces an $optimal$ rigid-body transformation T_C such that

$$T_C = \operatorname*{argmin}_{T \in \mathcal{T}} \sum_{k=1}^{m} D_{T(\mathbf{p_i})}(\mathbf{q}_{C[i]}) \tag{1}$$

We can now state the matching problem as follows:
Problem 1 (MATCH) *Given the above definitions, find the correspondence $C^\star$ and the optimal transformation induced by it, $T^\star$, such that*

$$\langle C^\star, T^\star \rangle = \operatorname*{argmin}_{C \in \mathcal{C}_{\mathcal{PQ}}, T \in \mathcal{T}} \sum_{i=1}^{m} D_{T(\mathbf{p_i})}(\mathbf{q}_{C[i]}) \tag{2}$$

Intuitively, a multipoint can be thought of as a model of a collection of points (e.g. atoms in an amino acid) that behaves as a rigid body i.e., the relative orientation of the points stays the same. Labels capture the intuition that an amino acid of feature X can only match another acid of the same feature. In simple cases, the feature could just indicate the residue type or the hydrophobicity of the amino acid. For larger motifs, the label could indicate the secondary structure (helix, loop, or strand) the amino acid is part of. Features can be aggregated ($\alpha = \{X, Y, Z\}$) for greater flexibility (α matches X, Y, or Z).

Thus, we have formulated MATCH as the problem of finding the optimal correspondence (and alignment) of the multipoints in the pattern set to the multipoints in the example set. The pattern set P and the example set Q are abstractions of a motif and a protein, respectively. Typically, each multipoint models one amino acid. This abstraction might depend only on the backbone atoms or on the side-chain atoms as well. Henceforth, we shall assume that each multipoint represents a list of points and the number (and the relative ordering) of these points is the same for two multipoints if they have the same label.

As it stands, however, the problem is too general. We need more information about how to model multipoints and their distance measures. By providing more information about these, we can formulate different special cases of the problem, each of which has a special biological relevance.
Problem 2 (COMPLETE-MATCH) *Solve MATCH under the following constraint: given any two multipoints $\mathbf{a}$ and $\mathbf{b}$, both of which represent sets of k points each, the distance between them is*

$$D^\circ_{\mathbf{a}}(\mathbf{b}) = \textstyle\sum_{i=0}^{k} ||\mathbf{a}(i) - \mathbf{b}(i)||^2$$

where $\mathbf{a}(i)$ is the i^{th} point in the list which the multipoint $\mathbf{a}$ represents and $||x - y||$ is the Euclidean distance between $x, y \in \Re^3$.
Problem 3 (PARTIAL-MATCH) *Solve MATCH under the following constraint: given any two multipoints $\mathbf{a}$ and $\mathbf{b}$, both of which represent sets of k points each, the distance between them is of the form*

$$D^+_{\mathbf{a}}(\mathbf{b}) = \textstyle\sum_{i=0}^{k} \rho(||\mathbf{a}(i) - \mathbf{b}(i)||)$$

where $\rho(x)$ is a monotonically non-decreasing function such that $0 \leq \frac{d\rho}{dx} \leq x$, and ρ is the same across all multipoints with a common label.

COMPLETE-MATCH abstracts the problem of looking for *well-preserved* matches of a given active site or sub-domain in a protein. PARTIAL-MATCH, on the other hand, would be of interest when we expect that the matching region (e.g., active site) may not be well-preserved. The distance measure used in COMPLETE-MATCH is equivalent to the Root Mean Squared Distance (RMSD) criteria. Under this distance criterion, it is relatively easy to find the optimal transformation $T^\star$ that minimizes $D^\circ_{T(\mathbf{a})}(\mathbf{b})$ [7,8]. The problem with this metric is that the influence of each pairwise distance is quadratic in the size of the distance. This creates problems when we expect a partial match between the two point-sets. In PARTIAL-MATCH, we use an *influence function*, $\rho(x)$, designed so that the larger inter-point distances do not overshadow the smaller ones. This idea is borrowed from the notion of *M-estimators* which are often used in Statistics and Computer Vision for robust estimation. The choice of a particular ρ depends on the situation and the biological motivation. For example, we might only be interested in matches where the distance between two matching points is less than a given threshold. The Tukey estimator captures that intuition. Similarly, this formulation can also be used in the case where the structure of the pattern (e.g., active site) is ambiguous.

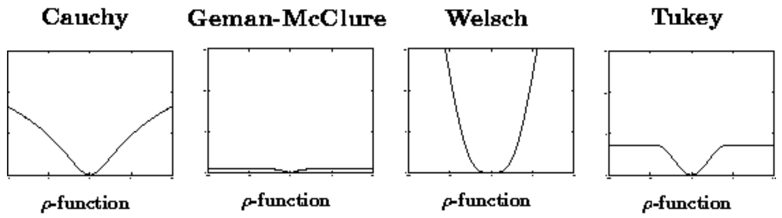


Figure 2: **M-estimators**

The above figure shows the plots of 4 important M-estimators: (a) Cauchy: $\rho(x) = \frac{c^2}{2} log(1 + (x/c)^2)$, (b) German-McClure: $\rho(x) = \frac{x^2/2}{1+x^2}$, (c) Welsch: $\rho(x) = c^2[1 - exp(-(x/c)^2)]$, (d) Tukey: $\rho(x) = \frac{c^2}{6}(1 - [1 - (x/c)^2]^3)$; where c is a constant.

2.1 Previous Work

The problem of identifying motifs in a protein has strong parallels to the object recognition problem in computer vision. Before we proceed, it is worth noting that many of the methods mentioned below rely on a well-known technique for the finding the optimal transformations— rotations and translations— for aligning two sets of n points i.e. the rotation and translation that result in the least squared-error in the alignment of the two point sets. The method, initially proposed by Faugeras [7] and Horn [8], assumes that the correspondences between the points in the two sets are known and boils down to finding the largest eigenvalue of a 4×4 matrix.

Kleywegt [9] et al. have released programs (SPASM, RIGOR) that look for small motifs (e.g., active sites) in a given protein using a brute-force approach. Their method, based on the idea that inter-point distances are invariant w.r.t. rotation and translation, performs an *exhaustive* search of sets of matching inter-point distances in the two point-sets. Their method can take advantage of the labeling information available with the motif and the protein and use it to prune the search space (e.g. a carbon atom should only match another carbon atom). Exhaustive searching, however, limits the size of motifs this method can be easily applied to.

Wolfson and Nussinov [10] proposed a technique (*Geometric Hashing*) also based upon the idea that distances are invariant under rotations and translations. An ordered triplet of points can be used to define a reference frame. For each ordered triplet of points in the point-set, coordinates of the other points (in the ref frame defined by the triplet) are stored in a hash-table. For points from the matching region, the key for the protein's hash-table will also be found in the motif's hash-table (and vice-versa) so that the appropriate transformation between the two ref frames (for the two triplets) can be found. The most popular transformation, across all triplets, wins.

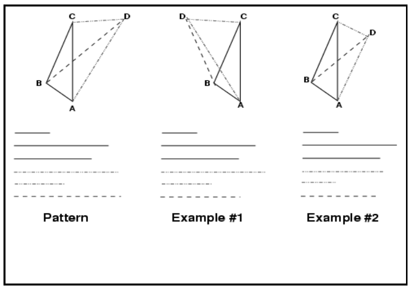
Geometric hashing is a powerful approach, but it has the same kind of problem as the approach of Kleywegt et al: it is hard to establish a link between the output of this algorithm and the desired optimum because they handle the matching problem in an indirect fashion. When two sets of n points match perfectly, the $\binom{n}{2}$ pairwise distances corresponding to each set will also match perfectly. However, when the match is imperfect, it can not be expressed easily in terms of an imperfect match between the corresponding pair-wise distances (see Fig 3). As such, it is not necessary that the final answer returned by this approach will be optimal in the LSE (least sum of squared errors) sense.

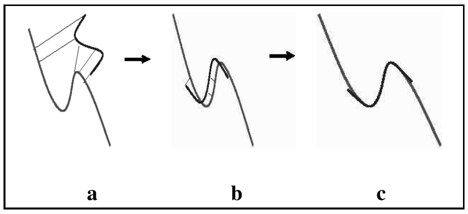
Iterative Closest Point Algorithm, proposed by Besl and McKay [11], computes a locally optimum match between two point sets i.e., it returns a correspondence and the induced optimal transformation between two unlabelled point sets. The distance metric is the same as the RMSD metric or the one used in COMPLETE-MATCH. The basic idea is that at any point, the (currently) best estimate of the optimal transformation can be used to improve the (currently) best estimate of the optimal set of correspondences and vice-versa (see Fig 4). By iterating over these operations, the method converges to a locally optimal solution. The algorithm is guaranteed to converge because in each step of the iteration, the least-mean-squared-error can only decrease. However, the minima which it reaches may well be a local minima and the algorithm is highly dependent on the starting placement for the pattern set w.r.t. to the example set.

3 Method

Here are some of the observations that guided our approach:

- In ICP, if we replace the Euclidean distance function by any other monotonically non-decreasing function of the Euclidean distance (see the defn of PARTIAL-MATCH), the algorithm is still guaranteed to converge to a local optimum because each step in each iteration of the algorithm can only reduce the error. Of course, the optimal transformation to map the pattern set to the example set must be optimal with respect to the specific distance measure and the methods of Faugeras et al may not be





3 4

Figure 3: **Matching Pairwise Distances Is Not Always a Good Idea**

The above figure shows a typical case where matching pairwise distances can lead to a sub-optimal result. Example #2 is more similar to the pattern than Example #1. However, only one line (BD) in Example #2 does not have the same length as the corresponding line in the pattern. In Example #2 there are three such lines (AD, BD, CD). Thus, Kleywegt's algorithm would choose Example #1 over Example #2.

Figure 4: **ICP**

The goal is to align the pattern (short) curve with the example (long) curve. (a): For each point of the pattern (short curve) set, find the closest point on the example (long curve) set and build the list of corresponding points. (b): Find the optimal transformation that aligns these corresponding points. (c): After many iterations the curves have been optimally aligned.

applicable.

- The quality of the final match returned by ICP depends on the initial placement of the pattern relative to the example. Suppose that we can identify, in the example, a few (small) *regions of interest* one of which, with high probability, also contains the optimal match. For each such region, we can *seed* ICP by placing the pattern near this region of the example.
- In our formulation, only multipoints of the same label can be matched. The proper choice of label for this purpose can help us in reaching the optimum quickly by reducing the size of the search space. The frequency of occurrence of different features among the set of amino acids is often uneven. For example, if one were to label amino acids just by their residue type, the most frequent amino acid, Leucine, is 6.85 times more abundant than the least frequent amino acid, Tryptophan[12].

Our method to solve MATCH is described in Algorithm 1. The initial step of our method consists of identifying regions of interest by using a small set of multipoints from the pattern as *pivots*. We find sets of multipoints from the example such that these multipoints could correspond to the pivoting multipoints in an optimal match. For each such possible correspondence, we transform the pattern so that the pivoting multipoints are aligned with their (guessed) counterparts. We then use an ICP-like algorithm to find the best match in that region. Two functions in the algorithm need further elaborations:

RegionsOfInterest This function returns the regions in the example set, Q, where a

Input: Given a pattern $P = \{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_m}\}$ and an example $Q = \{\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_n}\}$ where $\mathbf{p_i}, \mathbf{q_j} \in \mathcal{M}$ and $m \leq n$. Recall that $\mathcal{M}$ is the set of multipoints in $\Re^3$

Goal: Define the set of possible correspondences between P and Q: $\mathcal{C}_{PQ} = \{\langle r_1, r_2, \ldots, r_m\rangle | r_i \in \{1 \ldots n\},\ r_i \neq r_j \forall i, j \textit{ and } label(\mathbf{p_i}) = label(\mathbf{q}_{r_i})\}$. A transformation T is a rotation and translation in $\Re^3$. Since this operation is analogous for both points and multipoints, we shall use the symbol T for both the meanings, i.e., $T : \Re^3 \to \Re^3$ and also, $T : \mathcal{M} \to \mathcal{M}$. Find the optimal correspondence $C^\star \in \mathcal{C}_{PQ}$ and the corresponding transformation $T^\star$ such that

$$\langle C^\star, T^\star\rangle = \underset{\langle C,T\rangle}{\text{argmin}} \sum_{i=1}^{m} D_{T(\mathbf{p_i})}(\mathbf{q}_{C(i)})$$

Algorithm:

1. Generate seeds: choose some multipoints $\mathbf{p}_\alpha, \mathbf{p}_\beta, \ldots \in P$ such that $RegionsOfInterest((\mathbf{p}_\alpha, \mathbf{p}_\beta, \ldots), Q) = S$ returns only a few sets of possible matches.
2. Initialize $d_{best} = \infty$
3. For each $(\mathbf{q}'_\alpha, \mathbf{q}'_\beta, \ldots) \in S$, do
   (a) Find Initial Transform: $T_1 = OptimalTransform((\mathbf{p}_\alpha, \mathbf{p}_\beta, \ldots)\text{-}, (\mathbf{q}'_\alpha, \mathbf{q}'_\beta, \ldots))$
   (b) Initialize: $P_1[i] = T_1(P[i]), i = 1, \ldots, m$.
   (c) Initialize: $r = 1, d_0 = \infty, d_1 = LARGE\text{-}NUMBER$
   (d) While ($r < MAX\text{-}ITERATIONS$ and $d_r < d_{r-1}$)
      i. Set $r = r + 1$
      ii. Find correspondences: set correspondences so that each multipoint in P_r corresponds to the closest multipoint in Q:
      $$C_r[i] = \underset{y \in \{1,\ldots,n\}}{\text{argmin}}\ D_{P_{r-1}[i]}(Q[y])$$
      where $X[i]$ is the i^{th} multipoint in ordered set X
      iii. Find the optimal transformation for these correspondences: $T_r = OptimalTransformation((P[1], P[2], \ldots, P[m])\text{-}, (Q[C_r[1]], Q[C_r[2]], \ldots, Q[C_r[m]]))$
      iv. Set $P_r[i] = T_r(P[i])$, $i = 1, \ldots, m$
      v.
      $$d_r = \sum_{i=1}^{m} D_{P_r[i]}(Q[C_r[i]])$$
   (e) Update: if $d_{best} > d_r$ then set $C^\star = C_r$, $T^\star = T_r$, $d_{best} = d_r$
4. Return $C^\star, T^\star$

Algorithm 1 *General algorithm for solving MATCH* globally optimum match can be found. It takes as input a set of *pivoting* multipoints in the pattern P and returns a list of possibly optimal correspondences that this set can have in Q. This function can be implemented by choosing some multipoints from the pattern and looking, in the example set, for sets of multipoints whose labels and relative orientations are consistent with those of the chosen multipoints from the pattern. Our aim is to get only a few regions of interest. As the simplest choice, one could pick just one pivoting multipoint from the pattern and, from the example, choose all multipoints that have the same label as this multipoint. In a more sophisticated approach, one can pick a collection of multipoints from the pattern. The matching constraints would also be based on the relative orientation of these multipoints. Such constraints can be efficiently implemented, say, by using *kd*-trees.

OptimalTransform Given a set of corresponding multipoints $X = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}\}$ and $Y = \{\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_k}\}$, this function returns the transformation $T^\star$ that results in the minimum total error:

$$T^\star = \operatorname*{argmin}_T \sum_{i=1}^{k} D_{T(\mathbf{x_i})}(\mathbf{y_i})$$

In the case of COMPLETE-MATCH, the distance function is just the sum of the squares of the Euclidean distances between the points making up the two multipoints. As mentioned before, there are well known methods that can compute the optimal transformation (in a least squared error sense) such that final error is minimum [7,8]. In the case of PARTIAL-MATCH, our method is based on the following observation: the fraction of 'bad' data-points (or outliers), w, is expected to be low– otherwise the match won't have any biological significance. On picking a random sample from the data, there is a low probability that some the points in the sample will not be 'good'. This probability can be reduced further by taking samples many times. Then we can use the method by Faugeras and Horn, [7 8], to calculate an optimal transformation for this subset, *using the Euclidean distance measure*. Since our distance function gives lower weight to the bad data-points, this transformation should be a pretty good choice for the whole data too. Our method is similar to the RANSAC [13] method and the method of Venkatsubramanian et al [14,15]. Due to a lack of space, we can not provide a formal specification of our algorithm in this paper.

4 Analysis

The speed of Algorithm 1 depends on our ability to generate a small list of regions of interest in the example (protein) while ensuring that the global optimum will be one of these. There is a trade-off between the complexity of this step and the quality of its results. Still, the time it takes will, typically, be much smaller than the time taken by the rest of the algorithm.

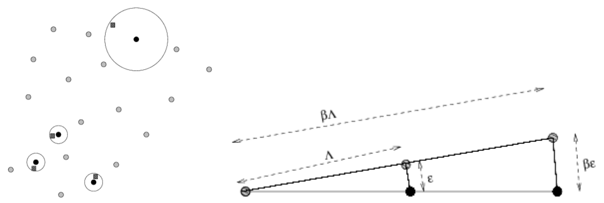
It should be clear that if a perfect match (exact replica of the pattern in the example) exists, Algorithm 1 will always detect it– one of the regions of interest will result in a perfect alignment of the pattern and the example.

In the second part of the algorithm, we iterate over each of the regions of interest discovered earlier and start our search by aligning the pivoting multipoints. The basic intuition is that once we have placed the pattern near the region of interest in the example, the algorithm can take us to the locally best match which could also be the global minimum. Of course, the algorithm might still go wrong and stray towards another local optimum even if the global optimum was indeed present in that region. We shall now try to provide some intuition for why something like this is unlikely i.e., once the algorithm has gotten on the trail towards the optimal match, it is unlikely that it will stray.

We analyze the case corresponding to COMPLETE-MATCH. Given a pattern set $\mathcal{U} \subset \Re^3$, we construct an example set $\mathcal{V} \subset \Re^3$ as follows: first, we apply a random rotation and translation to $\mathcal{U}$ and randomly add points around it such that the probability of a point being present in any given unit volume is γ. This is the test set $\mathcal{V}$. γ indicates the density of points in the test set– if the test set is densely packed, there should be a greater chance for any method to make mistakes, i.e., infer incorrect correspondences between points of the pattern set and those of the test set, even if it is close to the global optimum. Also, for notational convenience, we shall use $\mathcal{U}' \subset \mathcal{V}$ to refer to the subset of points which exactly match $\mathcal{U}$, i.e., we want the algorithm to output $\mathcal{U}[i] \leftrightarrow \mathcal{U}'[i]$ as the final correspondences. We also introduce the term β which is the ratio of the largest inter-point distance in $\mathcal{U}$ to the smallest inter-point distance in $\mathcal{U}$. β captures the shape of the pattern: skinny, cylindrical patterns will have a larger β than fat, cuboidal patterns.

Lemma 1 *If, at the end of Step* `3.(d).iii` *in Algorithm 1,* T_r *maps at least 3 points from* $\mathcal{U}$ *within an* ϵ*-neighborhood of their correct matches i.e.* $\|T_r(\mathcal{U}[s_i]) - \mathcal{U}'[s_i]\| < \epsilon$, $i = 1, 2, 3$, *then the probability that, for any point* $\mathcal{U}[i] \in \mathcal{U}$, $T_r(\mathcal{U}[i])$ *is not the closest point to* $\mathcal{U}'[i]$ *is less than* $\frac{4}{3}\pi\gamma(\beta\epsilon)^3$

Proof: Look at the Fig 5.



(a) (b)

Figure 5: **Algorithm 1 Won't Stray Too Often**

The filled squares represent points belonging to $\mathcal{U}$. The filled circles represent points belonging to $\mathcal{V}$. Of those, the lighter circles are randomly selected and the darker circles belong to $\mathcal{U}'$. If at least 3 points in $\mathcal{U}$ are within ϵ of the matching points from $\mathcal{U}'$, then any other point can only be $\beta\epsilon$ away from its corresponding point in $\mathcal{U}'$.

The probability that one or more randomly chosen points will be present within that neighborhood is less than $\frac{4}{3}\pi\gamma(\beta\epsilon)^3$ ■

5 Results

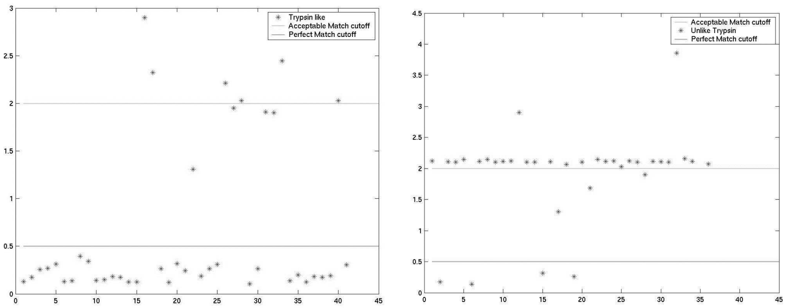
To test the quality of the matches returned by our algorithm (for COMPLETE-MATCH), we downloaded two sets of protein structures from the Protein Data Bank . We chose about 42 different variants of the protease trypsin and about 37 different types of kinases. Choosing the trypsin (consensus) active site [16] (3 amino acids) as our pattern, we ran our matching algorithm against the trypsins and then against the kinases.The results of our experiments are summarized in the plots shown below (Fig 6). For 32 of the 42 trypsin-like molecules, we were able to achieve matches with RMSDs less than 0.5 Å, indicating a near perfect match. In some other cases, the structural information in the PDB was incomplete and hence a good match could not be found. Finally, in the remaining 3 cases, our algorithm had not converged on to the optimal match.

When we tried to match the trypsin active site against the kinases, we expected to get low-quality matches. Most of the matches returned alignments where the RMSD was more than 2 Å– a low score considering that active site of trypsin is relatively small and hence a rough match for it can be found in many proteins. The interesting observation was that there were some kinases in which we could obtain really high quality matches. Some of these turned out to be buried inside the protein and hence could not be an active site. Such spurious matches, however, are a problem faced by almost all structure based techniques. Finally, there were 2 kinase-type molecules which matched the trypsin active site on their surface. It would be an interesting biological problem to determine if these kinases share any functional similarity with trypsins.

To evaluate **partial matches**, we distorted the trypsin active site by displacing two of the amino acids and then changing their orientation randomly. One of these amino acids was given a large displacement ($\simeq 8\AA$) while the other was given a smaller displacement ($\simeq 1.5\AA$). Our aim was to find an alignment that would *ignore the most outlying amino acid* and align (a part of) the protein with the rest of the motif. First, we used the same distance measure as in COMPLETE-MATCH i.e. the *Least-Sum-of-Squared-Error* criterion. We then replaced the the distance measure with one of the M-estimator (we used the Tukey Estimator) based criteria (PARTIAL-MATCH) and ran our algorithm. In one set of experiments, we used a gradient descent based approach to find the best transformation that minimized the error (in *OptimalTransform()*). In the other set of experiments, we used the method mentioned before (based on repeated sampling of a subset of points from the set). Fig 7 summarizes the results of a comparison of these three implementations with a distorted active site.

6 Conclusion

In this paper, we have formulated the problem of detecting motifs in proteins and have presented a general method for it. Our formulation of the problem provides for a rigorous measure of the best fit between a given pattern and an example. At the same time, there is a certain flexibility in the choice of an appropriate distance measure— biological context of the particular problem instance will be important in



(a) (b)

Figure 6: **COMPLETE-MATCH Results**
(a) Trypsin-like Proteins: **Y axis**: RMSDs between motif (trypsin active site) and (matching part of protein) in $\mathring{A}$. **X axis**: 42 proteins belonging to the Trypsin family. Observe that the quality of the match is near-perfect for most of the proteins. This confirms that all these proteins share the same structural and functional properties. (b) Kinase-like Proteins: **Y axis**: RMSDs between motif (trypsin active site) and (matching part of protein) in $\mathring{A}$. **X axis**: 37 proteins belonging to the Kinase family. Observe that most molecules don't have good matches.

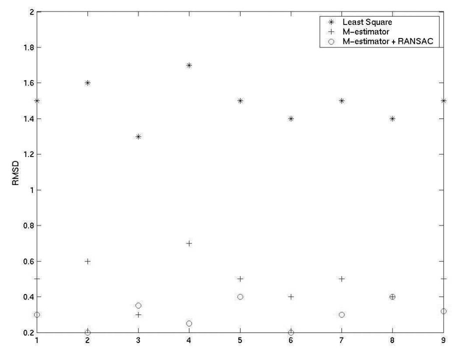


Figure 7: **Comparison of partial matching methods**
Y axis: RMSD between the aligned pattern and the matching sub-structure in the protein, *after removing the outlier pair from the match*. The pattern is a distorted version of the trypsin active site. **X axis** 9 different proteins from the Trypsin family. Here we compare the minimum alignment error after excluding the 'bad' amino acid. Ideally, this should be close to zero. Using the normal Least Square Error criterion gives the poorest performance. Using M-estimator based error criteria and gradient-descent based optimization technique, we can get better performance. If we use the RANSAC-like method mentioned before for finding the optimal transforms, we get the best results.

choosing this. Our method for solving the matching problem is fast and we also provided some intuition for why its results should be near-optimal most of the time. The algorithm, as presented, leaves significant scope in the choice of various parameters.

For example, the appropriate choice of features for labeling purposes will depend on the pattern/example at hand. In this context, we are in the process of conducting experiments to ascertain the performance of our algorithm under different choices of features. We are also testing our algorithm on much bigger data-sets and motifs of varying sizes. Another area that needs further work is related to finding the optimal transformation under PARTIAL-MATCH conditions. Though our current algorithm for *OptimalTransform()* works well, there is scope for improvement.

References

[1] K. Hofmann, P. Bucher, L. Falquet, A. Bairoch *The PROSITE database, its status in 1999* Nucleic Acids Res. 27:215-219, 1999

[2] A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S.R. Eddy, S. Griffiths-Jones, K.L. Howe, M. Marshall, and E.L.L. Sonnhammer *The Pfam protein families database* Nucleic Acids Research, 30(1):276-280, 2002

[3] L Lo Conte et al.*SCOP database in 2002: refinements accommodate structural genomics* Nucl. Acid Res. 30(1), 264-267, 2002.

[4] F.M.H. Pearl, et al. *Assigning genomic sequences to CATH* Nucleic Acids Research. Vol 28. No 1. 277-282, 2002

[5] M. Levitt and M. Gerstein *A Unified Statistical Framework for Sequence Comparison and Structure Comparison* Proc. Natl. Acad. Sci., 95, 5913-5920, 1998

[6] P. Bradley, P. S. Kim, B. Berger *Trilogy: Discovery of Sequence-Structure Patterns Across Diverse Proteins* International Conference on Research in Computational Biology (RECOMB), 2002

[7] O. D. Faugeras and M. Hebert, *The representation, recognition, and locating of 3-D objects*, Int. J. Robotic res. vol. 5, no. 3, pp. 27-52, Fall 1986.

[8] B. K. P. Horn, *Closed-form solution of absolute orientation using unit quaternions*, J. opt. Soc. Amer. A vol. 4, no. 4, pp. 629-642, Apr. 1987.

[9] GJ Kleyweg *Recognition of spatial motifs in protein structures* J Mol Biol. 29;285(4):1887-97, Jan 1999

[10] R. Nussinov, H.J. Wolfson *Efficient Detection of Three - Dimensional Motifs In Biological Macromolecules by Computer Vision Techniques*, Proceedings of the National Academy of Sciences, U.S.A., 88, 10495-10499, 1991

[11] P.J. Besl and N.D. Mckay, *A Method for Registration of 3-D Shapes*, IEEE Transactions on PAMI, Vol. 14. No.2, Feb. 1992.

[12] CRC Handbook of Chemistry and Physics (ISBN 0-8493-0458-X, CRC Press, Inc., Cleveland, Ohio)

[13] W. Forstner *Robust Estimation Procedures in Computer Vision* In Third Course in Digital Photogrammetry, 1998

[14] S. Venkatasubramanian *Geometric Shape Matching and Drug Design* Ph.D. Thesis, Stanford University, 1999

[15] P. Finn, L. Kavraki, R. Motwani, J.C. Latombe, C. Shelton, S. Venkatasubramanian and A. Yao *RAPID: Randomized Pharmacophore Identification in Drug Design* Proc. 13th ACM Symposium on Computational Geometry, 1997

[16] SF Russo, DN Morris *A fluorescent probe for the active site of bovine trypsin* Physiol Chem Phys Med NMR. 1983;15(3):223-7.