COMPUTATIONAL STUDY OF PROTEIN STRUCTURE AND FOLDING: SOME INTERESTING PROBLEMS

Rohit Singh

June 2002

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Master of Science.

Jean-Claude Latombe
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Master of Science.

Vijay Pande
(Dept of Chemistry)

Abstract

I am interested in the study of proteins: their structure and the way they fold into a stable structure. This is a vast field which requires collaboration between researchers from various disciplines. Computational study of protein structure and motion raises challenging problems of all kinds– from the basic and fundamental to the most applied. This report presents one of the interesting problems, idenftification of structural patterns in proteins, I have been working on, in collaboration with other researchers, at Stanford.

At the most basic level, an important question is: when are two protein fragments structurally similar ? This is hard to answer quantitatively because of the trade-offs between the size of the matching parts and the quality of the match. We study the problem of finding structural patterns (motifs) in a protein and present a rigorous, quantitatively-oriented formulation of the problem. We present a solution based on ideas from computer vision and statistics. We also show how to use the extra information available with proteins (e.g sequence information) in our framework. We present some theoretical justification for our methods and also discuss some of the experimental results. Part of this work was done jointly with Mitul Saha.

Acknowledgements

I am not the sole culprit behind the act of inflicting this report upon you. I had help; my family has been a co-conspirator all the way. My friends did redeem large parts of my soul by taking me to every movie, hike and road-trip they could think of. But in the end, they lost the battle, barely. The bulk of the blame for the current situation has to go the people who have taught me. At IIT Kanpur, I was heavily influenced by Prof Amitabha Mukerjee and Prof Manindra Aggarwal. Prof Mukerjee introduced me to the joys in doing useful things with computers, rather than just getting them to move bits around at ever faster speeds. Prof Aggarwal tried to teach me, somewhat unsuccessfully, the art of understanding the 'structure' of a problem before one went about proving theorems about it. I owe a lot of my enthusiasm for computer science to them.

At Stanford, I have had help from umpteen people. My advisor, Prof Jean-Claude Latombe, has been the most important influence– prodding, steering, teaching and peppering it all with nuggets of wisdom about life, research and everything in between. Working in an inter-discplinary field can have a lot of pitfalls and Prof Latombe also helped me find my bearings, teaching me how to separate the important from the not-so-worthwhile. I couldn't have done it without him. My group-mates and colleagues, some of the smartest people I have known, have taught me many things. I can not name all of them here, but I'd like to specially thank M. Serkan Apaydin, Itay Lotan, Mitul Saha, Fabian Schwarzer, Chris Varma, Bojan Zagrovic and Patrice Koehl. Prof Vijay Pande and Prof Michael Levitt often took the time to patiently explain a lot of the biological and physical context behind the problems. I thank them for all their patience. Finally, I would also like to thank Rajeev Singh and Satyajeet Salgar for

providing valuable comments on this report.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview and Motivation

1.1.1 The Bird's Eye View

Control of protein interactions (with each other and with DNA/RNA molecules) is the primary mechanism by which nature controls the functioning of living organisms. Typically, the interactions occur by a part of the protein docking into a complementary cavity on the same protein or some other protein. Alternatively, a small molecule (*ligand*) can bind into a complementarily-shaped cavity in a protein. Understanding and predicting protein structure, therefore, is an important biological problem. Among other things, this will lead to a better understanding and identification of biochemical pathways in the cell, help design newer approaches for intervening in malfunctioning systems (i.e. diseased tissues/organs) and help design and engineer new protein molecules. Understanding protein folding, i.e., how a chain of amino acids folds to attain its 3-dimensional structure will be very valuable in, say, nanotechnology where the ability of a system to self-assemble is critical.

Purely experimental strategies are often not sufficient for understanding protein structure and the protein folding process. One of the reasons is that experimental determination of structure is hard, time consuming and expensive. The SWISSPROT

database [1] contains sequence information about 109,000 proteins. However, information about protein structure is far less abundant. The Protein DataBank [22] contains structure information for only 18,000 proteins. While studying a protein, one often wants to understand the changes to the structure due to mutations in the genetic code or due to interactions with other molecules. This can be much easier to accomplish *in-silico* rather than *in-vivo*. The latter might involve a series of repetitive and time-consuming experiments.

Computational methods are also very valuable in studying protein folding problems. Understanding protein folding is important– proteins that don't fold well can cause problems (e.g. Alzheimer's disease or cystic fibrosis). Insights into the protein folding process can also help in determining ways to get protein-sized nanomachines to self-assemble into desired structures. At the same time, experimental determination of the intermediate conformations in a protein's folding pathway is very difficult because of the extremely short half-life periods of the intermediates. As such, computational studies of the folding process are often the only recourse left.

Computationally, a ball-and-stick model is typically the starting point for modeling and studying protein structure: the atoms are treated as balls and the bonds between them are treated as sticks (see Fig 1.1).

Using a knowledge of the distances between amino acids and the basic inter-atomic interactions (e.g. electrostatic, Van der Waals, hydrogen bonds) from chemistry an energy function could be computed for an arbitrary spatial arrangement of balls ('conformation'). Typically, there are constraints on the lengths of sticks (bond lengths) and mutual angles (bond angles). There is a lot of basic data, albeit imperfect, on the basic constraints and energy function. When predicting protein structures our goal is to identify the conformation that minimizes the energy and also satisfies the constraints. Similarly the problem of finding protein folding pathway(s) becomes the goal of identifying the sequence(s) of valid conformations leading from the unfolded state to the final, folded state and understanding the energetic/entropic properties of the intermediate conformations.

My work at Stanford has been in the area of computer modeling of protein structure and protein folding. This chapter provides a brief description of my work. The

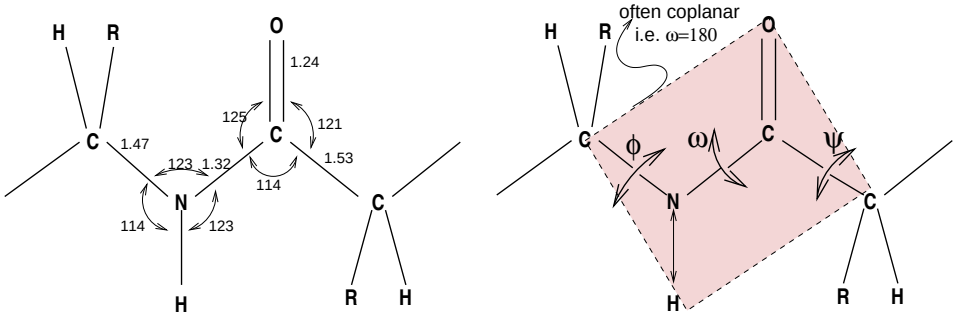


Figure 1.1: **Structure of the Peptide Bond**
Each amino acid consists of a C (commonly referred to as C_α) atom to which a NH_2, a COOH group and a side-chain R are attached. Amino acids differ only in the structure of their side-chains (e.g. R=H $\Rightarrow$ Glycine). Peptide bonds are formed by the combination of amide group (-NH_2) of one amino acid with the carboxyl group (-COOH) of another amino acid, forming the NH-CO bond (and one H_2O). (a) shows the bond lengths and bond angles involved. The lengths are in $\mathring{A}$ and the angles are in degrees. (b) shows the definitions of torsional angles in the backbone of a protein. It also indicates the planar geometry of the peptide bond.

next chapter describes my work in comparing protein structures in greater detail.

1.1.2 A Worm's Eye View

Zooming into this general framework, I will now describe the specific areas that I have been involved with and describe some of the related work being done at Stanford. Our group's (Prof. Jean-Claude Latombe's research group) work in studying protein modeling and protein behavior is split along two directions.

One is the study of *geometrical and kinematics issues involved in computational modeling of protein structures*. A particularly challenging problem has been to efficiently detect if there are collisions between the amino acids ("steric clashes") in a conformation (Lotan and Schwarzer [2]). This in an important step in studying protein structures, for it helps reduce the state-space (of valid conformations) by invalidating conformations that violate the given constraints. Lotan and Schwarzer have also been involved in developing efficient methods for calculation of approximate

Root Mean Squared Deviation metrics for measuring structural similarities between different conformations of a protein [4]. In this area, I have worked on the problem of geometric matching. In particular, we (*joint work with Mitul Saha*) are interested in finding (given) structural patterns/motifs inside protein structures. I describe, in this report, an efficient solution for searching for motifs in a database of protein structures. I have also been involved in determining efficient methods for solving the inverse kinematics problem when proteins are modeled in terms of their dihedral (torsional) angles.

The other direction of research in our group has been the study of protein folding– in particular, developing efficient methods for computation of ensemble properties in protein folding (transmission coefficients [5], [3]) and ligand-protein interaction (escape times [6]). My work, in this field, has been in collaboration with many people : M. Serkan Apaydin, Carlos Guestrin (Prof Latombe's group), David Hsu (Prof Jack Snoeyink's group, Univ of North Carolina at Chapel Hill) and others like Bojan Zagrovic, Chris Snow, Vijay Pande (Prof Vijay Pande's group). When studying a protein's folding pathways, one would like to estimate the probability of an arbitrary conformation of the protein folding into the stable conformation. The protein folding problem has parallels to the robot motion planning problem. I have been involved with other members of our group in exploring the application of Stochastic Roadmap Simulations (Serkan et al. [3]) for calculating the aforementioned probabilities.

1.2 Brief Description of My Work

1.2.1 Core Contribution

I will now provide a brief description of my work in the different areas I have mentioned before. The rest of this report will only focus, in greater detail, on the first part of the work described below.

1.2.2 Identifying Structural Motifs in Proteins

Structural patterns are often repeated across proteins. Thus, identifying these patterns (motifs) can give crucial insights into how a protein behaves– not only for small motifs but also for larger motifs. The search for smaller motifs (≤ 10 amino acids) is typically done to identify sites where a protein would interact with other molecules (*active sites*). Similar active sites would indicate similar functional characteristics. On the other hand, larger motifs often form the structural components of a protein and can also give insights into the functions of a protein. Finding a high-quality match between a motif and an arbitrary protein is hard because the match will almost always be imperfect and often partial. Current protein motif search algorithms are somewhat ad-hoc and lack rigor in quantifying the quality of match.

We first provide a formulation of the problem that is precise and rigorous. At the same time, the formulation is flexible enough to incorporate different match criteria. Then we pose this problem as an optimization problem. The crucial insight is that protein structures are much more than purely geometric entities– there is a considerable amount of information about amino acid sequences and secondary structure that can be used to prune the search space.

1.2.3 Representations and Algorithms for Modeling Proteins

Proteins can either be represented in terms of an all-atoms Cartesian coordinate representation or a (potentially) more compact representation based on the bond lengths, bond angles and dihedral angles. During the course of my research, I fiddled with a few different representation schemas for proteins and also experimented with different energy functions. While representing protein backbones with dihedral angles, it is common to assume that peptide bond has a planar geometry and hence the omega (ω) angle is 180°. Some of my experiments indicate this assumption is often violated. Similarly, the choice of an energy function is somewhat dependent on the choice of the modeling scheme. When representing a protein in terms of dihedral angles, an energy function based on a all-atoms representation can be unstable i.e. slight perturbations in the angles can produce vastly different energies.

When modeling proteins, using a representation scheme based on dihedral angles can lead to fewer redundancies. But many of the protein design goals are best expressed in Cartesian coordinates. In particular, we often need to solve the following problem: given an amino acid X at position A and another amino acid Y at position B, find a set of backbone dihedral angles for the protein such that protein forms a connected chain between this points. Additionally, if there are multiple sets of solutions there might be preferences for the positions of the amino acid residues in between. This problem has parallels to the inverse kinematics problem in robotics. The analogous problem there is to find the right orientation of links of a robot to get the end-effector of the robot at the desired point. But, currently, computational methods for solving this problem are known only for small chains (≤ 6 degree of freedom). In case of proteins there can be many more degrees of freedom. Since exact solutions for the general case are not known, one has to find approximate solutions.

My approach is to start with any legal conformation and then incrementally (but in a goal-driven way) deform the chain (to another legal conformation) until the conformation matches the constraints posed in the problem. In order to rapidly find the new conformation at every iteration, we use a sliding window based algorithm. The window is moved along the chain and the sub-chain inside the window is tweaked such that the points at the extremities of the sub-chain (*approximately*) retain their position. Since the window contains a small number of amino acids (i.e., only a few degrees of freedom), it is possible to apply inverse kinematic solution techniques from robotics to determine legal conformations for this sub-problem. In order to speed up each iteration, we maintain an efficient data structure (*kd-trees*) that stores precomputed solutions for chains of size of the sliding window. For the part of the chain inside the window, we can then query the data-structure to determine valid conformations for the amino acids within the window.

1.2.4 Stochastic Roadmap Simulations (SRS)

Monte Carlo techniques, in general, are a powerful way of simulating system's transitions from one state to another. For example, in order to determine the probability

of a protein changing its shape from one conformation to another (e.g. the folded state), Monte Carlo simulations can be applied. We start with a population of randomly chosen starting conformations. To each conformation, we apply a series of random, legal changes and observe how many members of the population reach the final state.

However, Monte Carlo simulation techniques are computationally intensive and can trace only one random walk at a time. In protein folding, a variety of pathways are possible and the probability of an event, summed across *all* pathways, could be of interest. Stochastic Roadmap Simulations (SRS), proposed by Serkan et al. [3], provides a way to study the protein folding problem and get a detailed understanding of all pathways and state transitions concurrently and at a much faster speed. The results are guaranteed to be asymptotically the same as those from Monte Carlo based techniques.

I explored the use of SRS in real-life cases, i.e., proteins of significant size/detail. In particular, I studied the folding of beta-hairpin and ColiE1 ROP protein (1ROP) using SRS. The former has been extensively studied by Zagrovic et al. [7] and thus comparative studies are possible.

1.3 Focus of This Report

In this report I restrict my discussion to only one of the problems I have mentioned above, the identification of structural motifs in proteins. Not only is the problem biologically important, it also poses interesting computational challenges. The next chapter discusses this problem and proposes a solution for it. I also discuss some applications of the proposed solution to biological problems. The final chapter provides a quick summary of my work.

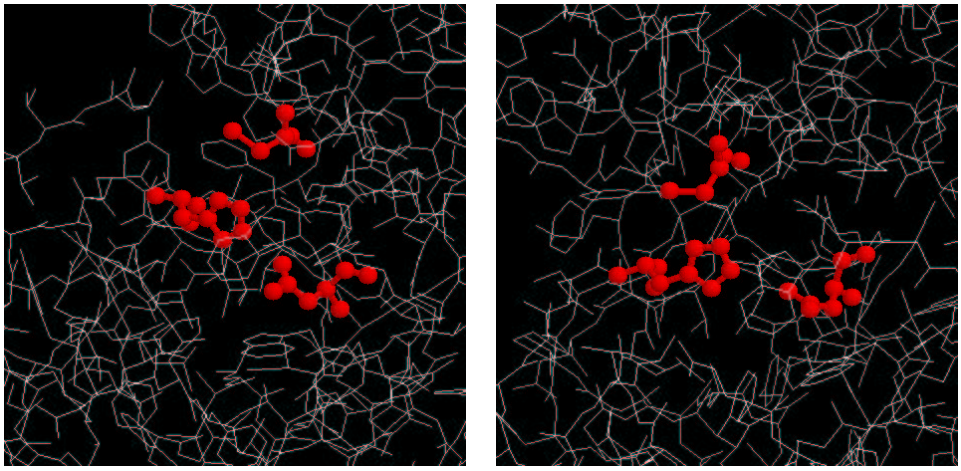
Chapter 2

Identifying Structural Patterns in Proteins

2.1 Motivation

One of the fundamental axioms of molecular biology is that three-dimensional structure governs function. Indeed, a similarity in function across different proteins can often be traced back to a similarity in structure. Thus, one can think of 3D structural patterns (*motifs*) that are often preserved through different protein structures (Fig 2.1, 2.2).

Such patterns can be considered as ordered sets of amino acids. However, unlike purely sequence based invariants, not only are the identities of the elements of the set (amino acids) preserved but their relative positions and orientations in 3D space are also (approximately) invariant. Active sites (Fig 2.1) and sub-domains (Fig 2.2) can be thought of as examples of such patterns. Active sites are patterns with relatively few (≤ 10) amino acids. The variations across different instances of the active-site are small and the active site is often the determining functional and structural feature of the protein. Sub-domains, in comparison, are much larger patterns, often consisting of a combination of a few secondary structure elements. The structural variations among different instances of a sub-domain can be relatively large. Large proteins are often made of a collection of such sub-domains. Classes of proteins sharing similar



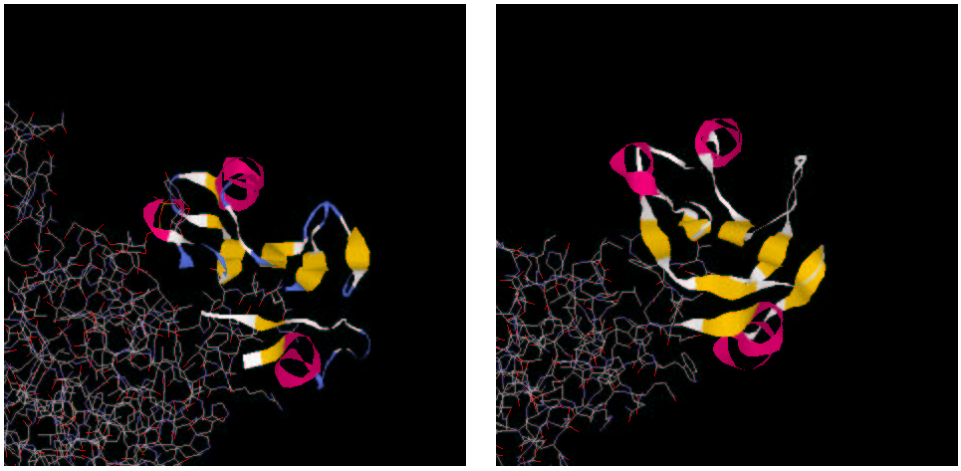
(a) (b)

Figure 2.1: **Example of a Small Motif**
These are active sites from PDB files 1PIP and 5PAD. Note that only 3 amino acids make the substructure of interest. Compare this with the much larger (and less conserved) motif of Fig 2.2

structural and functional properties often share a "signature" sub-domain. Indeed, the two most common methods of structure-based classification of proteins, SCOP and CATH, are based on this idea [14], [15].

A method to find a match for a given pattern in an arbitrary protein has numerous applications– protein-ligand binding, data-mining of databases of protein structures etc. Significant advancements in these low-level tasks will facilitate higher-level tasks like drug design. Most of the current approaches for the identification of active-sites rely on building sophisticated sequence-based models to build a "consensus" representation of the active site [12], [13]. However, these sequence based representations are only an approximation to the underlying structural information. A method based on structural matching would be more accurate.

The problem of finding a good match for a pattern in a protein has two parts. The first part is finding the best match for the pattern in the protein. The second part is to evaluate if this match is significant enough. For example, finding a close match for a pattern of 10 amino acids is more significant than finding a close match for a



(a) (b)

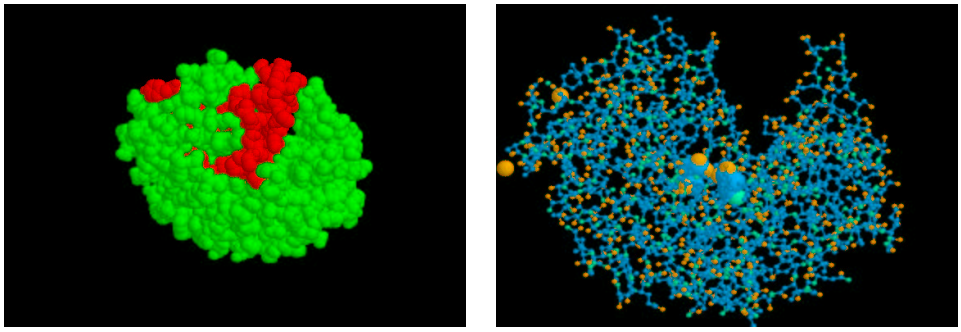
Figure 2.2: **Example of a Large Motif**
These are sub-domains, used by SCOP [14] for classification, that have been taken from the PDB files 1E8C and 1GG4. Compare this figure with Fig 2.1 and note the difference in the size of the motifs. Also, observe that the active site of Fig 2.1 is better conserved across the two proteins.

pattern for 2 amino acids. The process of evaluating the quality of a match, however, is somewhat subjective, depending on the goal of the biologist. There has not been a lot of work on it, but we refer the interested reader to [21], [16]. 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 given structural pattern in a given molecule. We then discuss a method for solving the problem and evaluate it.

2.2 Problem Formulation

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*



(a) (b)

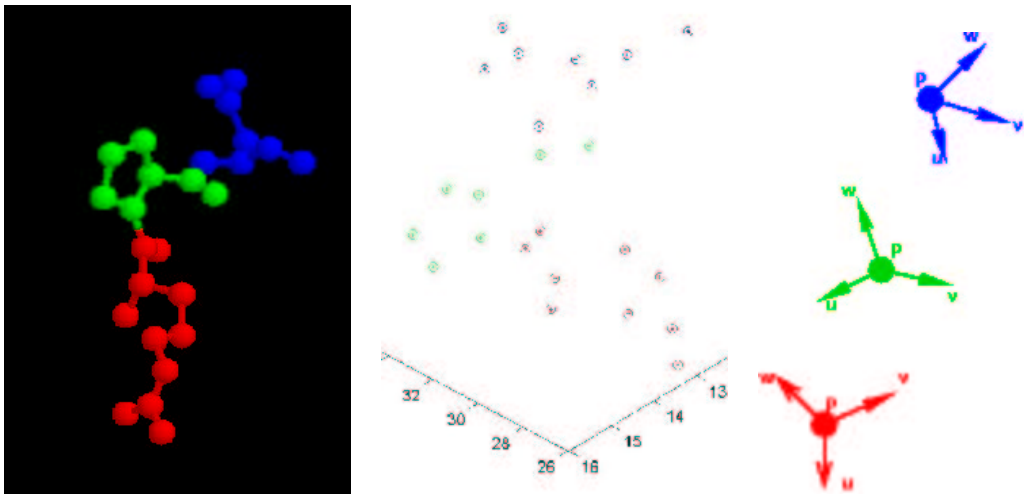
Figure 2.3: **Example of an Active Site**
The above figures show the active site of pepsin, a digestive protein. (a) Pepsin is the green colored molecule. Notice the V-shaped groove where a red colored molecule is fitted. This molecule acts as a pepsin inhibitor and prevents pepsin from acting on the body's own proteins. (b) This is a ball-and-stick view of the pepsin molecule, without the inhibitor. Notice the V-shaped groove. At the bottom of the groove are two amino acids (enlarged for emphasis) which are instrumental in cleaving a protein chain. A lot of digestive enzymes share this kind of an active site. One of our sub-goals is to be able to recognize, given the active site of, say, pepsin, similar active sites in other proteins too.

$$\mathbf{a} = \langle \vec{p}, \langle \hat{u}, \hat{v}, \hat{w} \rangle \rangle$$

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* $\vec{p}$ *and* $\mathcal{M}$ *is set of all multipoints. Also, every multipoint* $\mathbf{a}$ *has a label* $l_{\mathbf{a}} \in \mathcal{L}$ *where* $\mathcal{L}$ *is the set of all possible labels.*

We shall call $\vec{p}$ the *anchor* of $\mathbf{a}$. Optionally, a distance measure between multipoints of the same label can be defined. A multipoint $\mathbf{a} \in \mathcal{M}$ can have an associated distance measure $D_{\mathbf{a}}(\mathbf{b})$ which provides its distance from some other multipoint $\mathbf{b}$ which has the *same label* as $\mathbf{a}$. The calculation of this distance may depend on the internal representation of the two multipoints.

Intuitively, a multipoint can be thought of as a model of a collection of points (e.g. atoms in an amino acid) whose relative orientation stays the same even though the whole group can undergo a rotation and/or translation. Associating a reference



(a) (b) (c)

Figure 2.4: **Multipoints**
The above figures show how a multipoint would typically be constructed. A protein structure (a) can be broken into a collection of 'significant' points (b). For each group of points that belong to one feature, we can construct an anchor and a reference frame (c).

frame with each multipoint enables one to define a rigid-body transformation that maps one multipoint into another. The distance measure between two multipoints will depend on the distances between the two sets of points they model. Given a multipoint $\mathbf{a} = \langle \vec{p}, \langle \hat{u}, \hat{v}, \hat{w} \rangle \rangle$, a rigid-body transformation $T \in \mathcal{T}$ is defined so that applying T on $\mathbf{a}$ is the same as applying T on the points which $\mathbf{a}$ represents: $T(\mathbf{a}) = \mathbf{a}' = \langle T(\vec{p}), \langle T(\hat{u}), T(\hat{v}), T(\hat{w}) \rangle \rangle$. The distance measure associated with $\mathbf{a}$, if any, will be preserved i.e. $D_{\mathbf{a}}(\mathbf{b}) = D_{T(\mathbf{a})}(T(\mathbf{b}))$. $\mathcal{T}$ is the set of all rigid-body transformations in $\Re^3$ i.e. some combination of translation(s) and rotation(s). One convenient way of representing such a transformation would involve 7 variables: 3 variables for an arbitrary translation and 4 variables for a quaternions based representation of an arbitrary rotation [17]. The motivation behind associating a distance measure with each multipoint individually will become clear shortly.

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. These multipoints need not have distance measures associated with them. 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

$$T_C = \min_{T \in \mathcal{T}} \sum_{k=1}^{m} D_{T(\mathbf{p_i})}(\mathbf{q}_{C[i]}) \tag{2.1}$$

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*

$$\langle C^\star, T^\star \rangle = \operatorname*{argmin}_{C \in \mathcal{C}_{\mathcal{PQ}}, T \in \mathcal{T}} \sum_{i=1}^{m} D_{T(\mathbf{p_i})}(\mathbf{q}_{C[i]}) \tag{2.2}$$

In the definition of a multipoint, labels capture the intuition that an amino acid of feature X can only match another acid of the same feature. In the simplest case, the feature could be just the (residue) type of the amino acid. It could also indicate if the amino acid is hydrophobic or polar. For larger motifs, the label could indicate the secondary structure (helix, loop or strand) the amino acid is part of. If we want to express the condition that feature X can match features X, Y or Z, we can just introduce an *aggregate* feature α and replace all X, Y or Z by α.

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 is an abstraction of a structural motif. The example set Q abstracts the protein we are querying. Typically, each multipoint models one amino acid. In particular, we might choose 3 points from the 'backbone' of the amino acid (the N, C_α and C' atoms, see Fig 1.1). Otherwise, one or more of the points may be from the side-chain of the amino acid. Regardless of the way

these points are chosen, this collection of points can be used to generate an anchor and a reference frame, thus producing a multipoint. Henceforth, we shall assume that each multipoint represents a list of points and the number of these points is the same for two multipoints if they have the same label. Moreover, multipoints with the same label should have the same structure, i.e., the way in which the anchor and the reference frame were generated from the collection of points should be the same for all multipoints sharing a common label.

However, as it stands, the problem is too general. We need more information about how to model multipoints and their distance measures. By providing more information about these can formulate different special cases of the problem, each of which has a special biological relevance.

COMPLETE-MATCH

Here, the distance between two multipoints is the sum of the squares of the Euclidean distance between the points making up the multipoint.

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*

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

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* $x, y \in \Re^3$*.*

Note that this measure of distance directly corresponds to the Root Mean Squared Distance (RMSD) metric which is commonly used in biology. Moreover, as we shall soon see, it is relatively easy to find the optimal transformation $T^{\star}$ that minimizes $D^{\circ}_{T(\mathbf{a})}(\mathbf{b})$. The problem with this metric is that the influence of each pairwise distance is quadratic in the size of the distance. This raises problems when we expect a partial match between the two point-sets.

PARTIAL-MATCH

We often need to capture the intuition that it is better to have a good fit on a subset of the pattern point-set rather than a mediocre fit on all the points in the pattern. For this purpose, we can borrow the idea behind *M-estimators*[27]. M-estimators are statistical techniques designed for estimating statistical parameters in the presence of outliers. We define $D^+_{\mathbf{a}}()$ so that its dependence on the pairwise distances between points is *not* quadratic. Instead, we use an *influence function* that grows more slowly:

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*

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

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.*

Thus, $\rho(x)$ is an *influence function* designed so that the larger inter-point distances do not overshadow the smaller one. The choice of a particular ρ depends on the situation and the biological motivation. Some of the more commonly used choices are shown in Fig 2.5. Some of these influence functions are rather intuitive. For example, we might only be interested in matches where the distance between two points is less than a given threshold. The Tukey estimator captures that intuition.

IMPRECISE-MATCH

Often, only imprecise information is available about the positions of the multipoints in the pattern set. It would be preferable to capture this ambiguity in the distance function. This can be done by modifying the construction of $D^+_{\mathbf{a}}(\cdot)$ from PARTIAL-MATCH. For example, if it is expected that a point could lie within d units of a central position, we might design ρ to reflect that:

$$\rho(x) = \begin{cases} 0 & \text{if x<d} \\ (x-d)^2 & \text{otherwise} \end{cases}$$

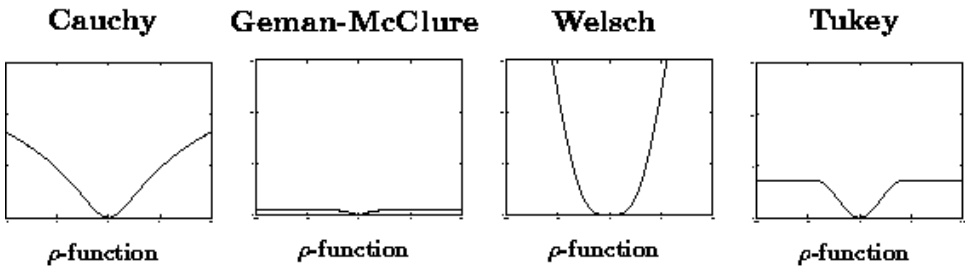


Figure 2.5: **M-estimators**
The above figure shows the plots of 4 important M-estimators: a) Cauchy: $\rho(x) = \frac{c^2}{2} log(1 + (x/c)^2)$, b) German-McClure: $\rho(x) = \frac{x^2/2}{1+x^2}$, c) Welsch: $\rho(x) = c^2[1 - exp(-(x/c)^2)]$, d) Tukey: $\rho(x) = \frac{c^2}{6}(1 - [1 - (x/c)^2]^3)$; where c is a constant.

Note that the specification of IMPRECISE-MATCH and PARTIAL-MATCH are the same, so that an algorithm to solve PARTIAL-MATCH could also be used for IMPRECISE-MATCH. In the rest of this discussion, we will not be discussing IMPRECISE-MATCH specifically.

It should be obvious that each of these 'theoretical' formulations is an abstraction of a corresponding biological problem. 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 will not be well-preserved and some of the amino acids part of, say, the active site could be significantly displaced from their expected (relative) position. Lastly, IMPRECISE-MATCH models that situation.when we observe a lot of variation in the structure of an active site across different proteins and need to encode that ambiguity in our problem description,

2.3 Related Work

The identification of a structural pattern in a protein has strong parallels to the object recognition problem in computer vision.

Before we proceed, its worth noting that many of the methods mentioned below rely on a well-known technique for the finding the optimal rotation and translation

for aligning two sets of n points, i.e., the rotation and translation that results in the minimum RMSD in the alignment of the two point sets. The method, initially proposed by Faugeras [9] and Horn [10], assumes that the correspondences between the points in the sets are known and boils down to finding the largest eigenvalue of a 4×4 matrix.

Kleywegt et al.[30] have released programs (SPASM, RIGOR) that look for small motifs (e.g. active sites) in a given protein using a brute-force approach. Their methods are based on the idea that inter-point distances are invariant w.r.t. rotation and translation. They start by modeling the motif and the protein as sets of points in 3D. Then they calculate pair-wise distances between points of the motif and repeat this process for the points of the protein. Since distances are invariant w.r.t. rotations and translations, we only need to find a set of (self-consistent) inter-point distances in the protein that match the corresponding inter-point distances in the motif. Thus, we can infer the correspondences between the points in the pattern (motif) and the example (protein). Once the correspondences have been found, it is easy to solve for best possible transformation (rotations and translations only) which map the points of the pattern to the corresponding points in the example (assuming that we are solving COMPLETE-MATCH). Thus, one only needs to do an *exhaustive* search in the space of inter-point distances. Their approach also uses the labeling information available with the motif and the protein and uses it to prune the search space (e.g. a carbon atom should only match another carbon atom).

The basic problem with this method is that it handles 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. 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 (see Fig 2.6). The other problem with this approach is that because it performs an exhaustive search of the possible correspondences, it gets infeasible as soon as the size of the motif increases beyond a few amino acids.

In 1991, Wolfson and Nussinov [32] proposed a technique (*Geometric Hashing*) also

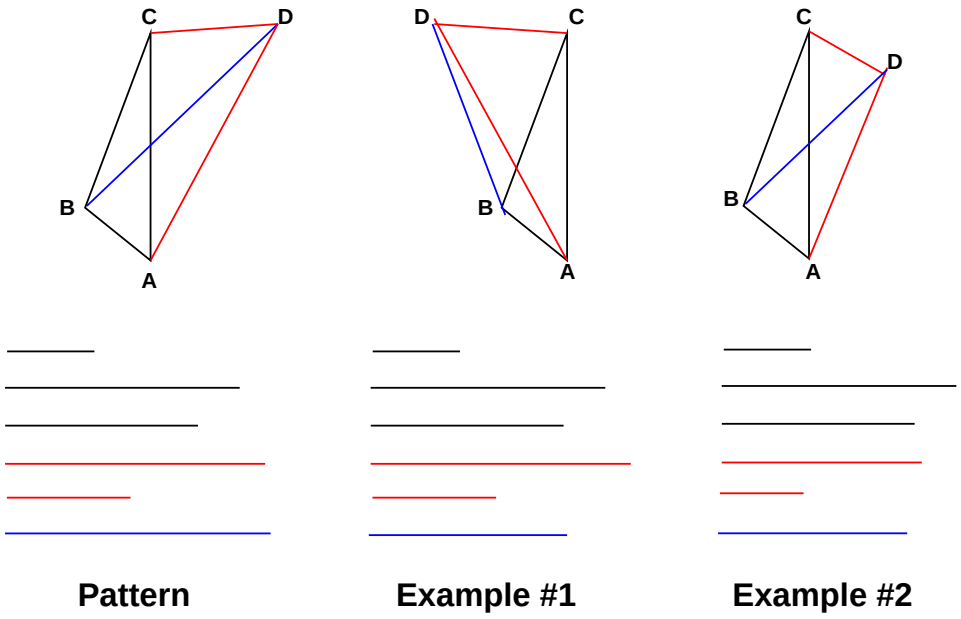


Figure 2.6: **Matching Pairwise Distances Is Not Always a Good Idea**
The above figure shows a typical case where matching pairwise distances can lead to a sub-optimal result. Example #2 is more similar to the pattern than example #1. However, only one line (the blue line) in Example #2 does not have the same length as the corresponding line in the pattern. In example #3 there are three such lines (all the red and blue lines). Thus, Kleywegt's algorithm would choose example #1 over example #2.

based upon the idea that distances are invariant under rotations and translations. This involves a preprocessing step. For each triplet of points in the example set (protein) they calculate the position of all other points w.r.t. the reference frame defined by these points. Then they build a hash table where the key describes the relative orientation of the three points w.r.t. each other and the relative orientation of a point in the example set w.r.t. this triplet. The value corresponding to this key contains the actual indices of the relevant points. Given a pattern set (motif) the same procedure is repeated. For each key in the motif's hash table, we can then use it to index into the protein's hash table and find the corresponding sets of points. We then calculate the optimal set of transformations for each such correspondence (assuming the minimum RMSD criteria). This transformation receives votes from each point in the pattern depending upon how happy the point is with the transformation– i.e.

how well the point is matched. The most highly voted transformation wins. This idea and its variants are parts of several techniques [33], [34]. Pennec et al proposed a modification in which they exploited the geometry of amino acid structure to attach a reference frame to each amino acid than triplets of amino acids. This reduced the size of hash-table significantly. 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. Also, this method requires significant amounts of preprocessing which can be undesirable in certain situations.

Venkatasubramanian et al. [35], [36] have worked on the problem of discovering structural invariants in a collection of different conformations of the same molecule. They formulate the problem in terms of finding a common substructure of size $\geq \alpha n$ in a point-set of n points and present randomized algorithms for this problem. Some of the intuition behind their algorithms is also shared by Algorithm 3 presented later in this report. However, they are primarily concerned with discovering only well-conserved invariants across point-sets of the same size. Our algorithms are oriented towards the case where the example would typically be much larger than the pattern, leading to many regions of interest in the example. Also the condition that the pattern be well-conserved in the example is often violated in our problems. Indeed, our goal is to let the biologist specify, by defining an appropriate influence function $\rho(x)$, the acceptable trade-off between size of the match and the quality of the match.

2.3.1 Iterative Closest Point

In 1992 Besl and McKay [28] presented the Iterative Closest Point Algorithm. The algorithm provides 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. By iterating over these

operations, the method converges to a locally optimal solution. Algorithm 1 describes their method in greater detail.

The algorithm is guaranteed to converge because in each step in Algorithm 2, the least mean squared error can only decrease i.e. $d_r \leq d_{r-1}$. However, the minima which it reaches may well be a local minima and the algorithm is highly dependent on the the starting position for the data set w.r.t. to the model set.

2.4 Method

Just to recapitulate, we are interested in the problem of finding an optimal match between two sets of multipoints (MATCH). Each multipoint represents a collection of points and only multipoints with same labels can match. The distance measure between two multipoints can be an arbitrary function.

Here are some of the observations that guided our approach:

- In Algorithm 1 (ICP), if we replace the Euclidean distance function by any other function, 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 i.e. $d_{r+1} \leq d_r$. Of course, the optimal transformation to map $\mathcal{S}_r$ to $\mathcal{Z}$ must be optimal with respect to the specific distance measure and the methods of Faugeras et al. may not be 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 also contains the optimal match. For each such region, we can seed ICP by placing the pattern near this region of the example.

- 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 [39]. In our formulation, only multipoints of the same label can be matched. The proper choice of label

Input: Given a pattern set (motif) $\mathcal{S} = \{s_1, s_2, \ldots, s_J\} \subset \Re^3$ and an example set (protein) $\mathcal{Z} = \{z_1, z_2, \ldots, z_K\} \subset \Re^3$, $J \leq K$, we can define a correspondence $C : \{1, \ldots, J\} \rightarrow \{1, \ldots, K\}$ as an injection (one-to-one mapping) from the points in $\mathcal{S}$ to the points in $\mathcal{Z}$. A transformation $T : \Re^3 \rightarrow \Re^3$ is a rotation and translation of points in $\Re^3$.

Goal: Find the optimal transformation $T^\star$ and the optimal correspondence $C^\star$ such that

$$\langle C^\star, T^\star \rangle = \underset{\langle C,T \rangle}{\operatorname{argmin}} \sum_{i=1}^{J} \|T(s_i) - z_{C(i)}\|^2$$

Algorithm:

1. Initialize: $\mathcal{S}_0 = \mathcal{S}, r = 1, d_0 = \infty$
2. While ($r < MAX\text{-}ITERATIONS$ and $d_{r-1} > ERROR\text{-}THRESHOLD$)
   (a) Set correspondences so that each point in $\mathcal{S}_r$ corresponds to closest point in $\mathcal{Z}$:
   $$C_r(i) = \underset{y \in \{1,\ldots,K\}}{\operatorname{argmin}} \|\mathcal{S}_{r-1}[i] - \mathcal{Z}[y]\|$$
   where $\mathcal{S}[i]$ is the i^{th} point in $\mathcal{S}$
   (b) Find the optimal transformation for these correspondences using the method by Faugeras and Horn, [9] [10]:
   $$T_r = \underset{T}{\operatorname{argmin}} \sum_{i=1}^{J} \|T(\mathcal{S}_0[i]) - \mathcal{Z}[C_r(i)]\|^2$$
   (c) Set $\mathcal{S}_r[i] = T_r(\mathcal{S}_0[i])$, $i = 1, \ldots, J$
   (d)
   $$d_r = \sum_{i=1}^{J} \|\mathcal{S}_r[i]) - \mathcal{Z}[C_r(i)]\|^2$$
   (e) Set $r = r + 1$
3. Return $C^\star = C_{r-1}, T^\star = T_{r-1}$

Algorithm 1 *Iterative Closest Point Algorithm*

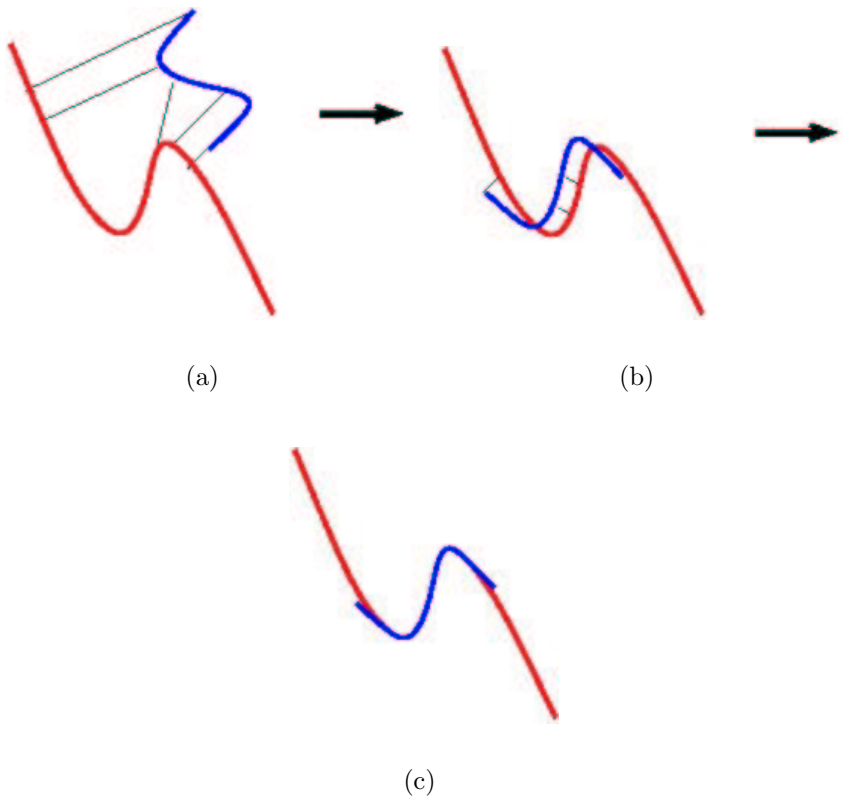


Figure 2.7: **ICP**
The goal is to align the red (short) curve with the blue (long) curve. (a): For each point of the pattern (red) set, find the closest point on the example (blue) set and build the list of corresponding points. (b): Find the optimal transformation that aligns these corresponding points. (c): After many iterations the curves have been optimally aligned.

for this purpose can help us in reaching the optimum quickly by reducing the number of regions chosen.

Our method to solve MATCH is described in Algorithm 2. 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 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.

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: $\mathcal{C}_{PQ} = \{\langle r_1, r_2, \ldots, r_m\rangle | r_i \in \{1 \ldots n\},\ r_i \neq r_j \forall i, j \text{ and } label(\mathbf{p_i}) = label(\mathbf{q}_{r_i})\}$. 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., $T : \Re^3 \rightarrow \Re^3$ and also, $T : \mathcal{M} \rightarrow \mathcal{M}$. Find the optimal correspondence $C^\star \in \mathcal{C}_{PQ}$ and the corresponding transformation $T^\star$ such that

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

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: $T_1 = OptimalTransform((\mathbf{p}_\alpha, \mathbf{p}_\beta, \ldots), (\mathbf{q}'_\alpha, \mathbf{q}'_\beta, \ldots))$
   (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:
      $$C_r[i] = \underset{y \in \{1,\ldots,n\}}{\operatorname{argmin}} D_{P_{r-1}[i]}(Q[y])$$
      where $X[i]$ is the i^{th} multipoint in ordered set X
      iii. Find the optimal transformation for these correspondences: $T_r = OptimalTransformation((P[1], P[2], \ldots, P[m]), (Q[C_r[1]], Q[C_r[2]], \ldots, Q[C_r[m]]))$
      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 2 *General algorithm for solving MATCH*

Two functions in the algorithm need further elaborations:

RegionsOfInterest This function returns the regions in the example set, Q, where a globally optimum match can be found. It takes as input a set of *pivoting* multipoints $(\mathbf{p}_\alpha, \mathbf{p}_\beta, \ldots)$ in the pattern P and returns a list of possibly optimal correspondences that this set can have in Q. Each set in this list, $(\mathbf{q}'_\alpha, \mathbf{q}'_\beta, \ldots)$, indicates that the correspondences $\mathbf{p}_\alpha \leftrightarrow \mathbf{q}'_\alpha, \mathbf{p}_\beta \leftrightarrow \mathbf{q}'_\beta, \ldots$ can be part of a globally optimum match. Thus, at the start of each iteration in Step 3 we have a new region of interest to explore. Step 3.(a) transforms the pattern P so that the pivoting multipoints are aligned with the corresponding multipoints from 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. While looking for sets of matching multipoints in the example, the constraints would also be based on the relative orientation of these multipoints. Such constraints can be efficiently implemented, say, by using *kd*-trees. So if our active-site has a Tryptophan and a Lysine within 5 $\mathring{A}$ of each other, we could search in the protein for those areas which have a Tryptophan and a Lysine within, say, 7 $\mathring{A}$ of each other. The best choice of pivoting multipoints from P and the related constraints will require some biological knowledge. For example, if we chose only one pivoting multipoint from P, it is best to choose one with a label that is expected to be the least frequent in the example set (e.g. prefer Tryptophan over Leucine).

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:

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

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. These methods rely on computing a 4×4 matrix from the given data and calculating the largest eigenvalue of this matrix [9], [10].

In the case of PARTIAL-MATCH, *OptimalTransform()* will reduce to finding the optimal transformation under a distance measure $D^{+}()$ (see Problem 3). I am not aware of any methods that can provide a provably optimal transformation for an arbitrary choice of $\rho()$ in such a distance measure. One approach is to express the transformation in terms of some variables, (e.g. 7 variables in a quaternion-based representation), and use some optimization technique (e.g. gradient descent) to search for the optimum solution in this space. But such methods are slow and can get stuck in local minima.

We have built our method for finding the optimal transformation 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, [9] [10], to calculate an optimal transformation for this subset, *using the Euclidean distance measure*. Since our distance function gives lower weight to the bad cases, this transformation should be a pretty good choice for the whole data too. Our method, described in Algorithm 3, is similar to the RANSAC method [43]. In Algorithm 3, the (arbitrary) distance measure is used to estimate the error threshold beyond which a matching point should be classified as

an outlier. The best transformation from each region of interest (Step 2, Algorithm 2) is scored using the (arbitrary) distance measure (Step 2.v, Algorithm 2) and the transformation with the lowest error under this distance measure is chosen.

2.5 Analysis

The efficacy of Algorithm 2 depends on two factors. One is 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. The second factor is the ability of the algorithm to explore any region of interest efficiently. We will discuss these in slightly greater detail now.

The first step of Algorithm 2 is to pick some multipoints from the pattern and find all sets of multipoints from the example that are consistent with this set. This step should execute reasonably quickly and output only a small set of possible matches (yes, there is a trade-off). The running time of this step will depend on the constraints and the number of pivoting multipoints chosen, but it is expected to be much less than than the running time of Step 3.

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 chosen multipoints. Note that every multipoint has a reference frame so that it is possible to find a transformation even with just one pair of matching multipoints. The basic intuition is that once we have place 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 a sub-optimal match 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 a test 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

Input: We are given two sets of points in 3D: $X, Y \subset \Re^3$, $|X| = |Y| = k$. We also know the correspondences between these points i.e. $X[i]$ corresponds to $Y[i]$. We are also given a distance function between these points, $D_X^+(Y) = \sum_{i=1}^{k} \rho(\|X[i] - Y[i]\|)$. We also have prior knowledge that the fraction of outliers in X is expected to be less than w, $0 < w < 1$.

Goal: Find a transformation T that minimizes $D_{T(X)}^+(Y)$.

Algorithm:

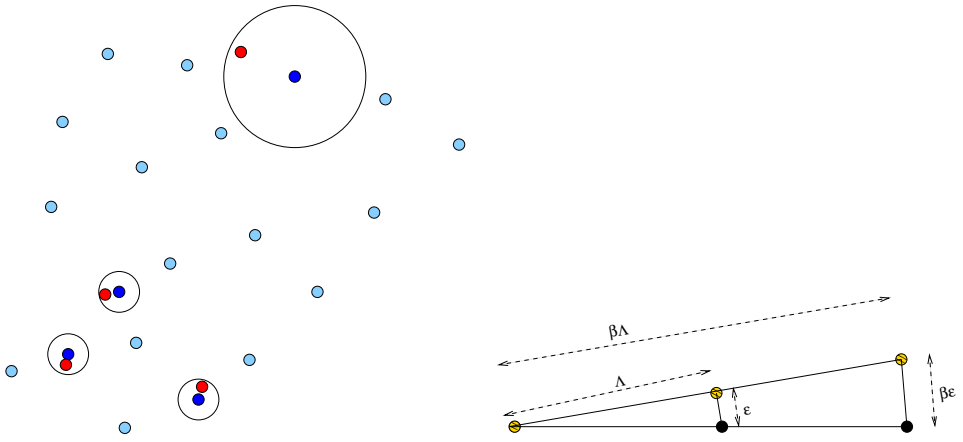
1. Initialize r to be some small number greater than 2. This is the size of sample we will draw each time.
2. Choose $ERROR\text{-}THRESHOLD$ such that if $\|p_{actual} - p_{expected}\| > ERROR\text{-}THRESHOLD$ then p_{actual} is classified as an outlier (according to $D_X^+()$).
3. Choose d such that $d > \log(1-\alpha)/\log(w)$ where α indicates the required confidence level in the final output (e.g. $\alpha = 0.95$). Set $n_{best} = 0$.
4. For $i = 1, \ldots, cw^r \log(n)$, where c is some constant, do:
   (a) Pick a sample $s_i \subset \{1, \ldots, k\}$, $|s_i| = r$, randomly. Denote $A_{s_i} = \{A[s_i[1]], \ldots, A[s_i[r]]\}$, where $A[u]$ is the u^{th} element in A.
   (b) Using the method of Faugeras and Horn [9], find the transformation T_i such that $T_i = \underset{T}{\operatorname{argmin}} \sum_{l=1}^{r} \|T(X_{s_i}[l]) - Y_{s_i}[l]\|^2$
   (c) Apply T_i on all points in X and find the points that agree with this transformation: $X_{T_i} = \{l | 1 \le l \le k, \|T_i(X[l]) - Y[l]\| < ERROR\text{-}THRESHOLD\}$. Set $n_i = |X_{T_i}|$.
   (d) If $n_i > n_{best}$, $T_{best} = \underset{T}{\operatorname{argmin}} \sum_{l \in X_{T_i}} \|T(X_{s_i}[l]) - Y_{s_i}[l]\|^2$
   (e) if $n_i \ge d$, **break**
5. Return T_{best}

Algorithm 3 *OptimalTransform() for PARTIAL-MATCH*

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 2, 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 2.4. The red points belong to $\mathcal{U}$. The blue points belong to $\mathcal{V}$.



(a) (b)

Figure 2.8: **Algorithm 2 Won't Stray Too Often**

Of those, the light blue points are randomly selected and the dark blue points 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 $\frac{4}{3}\pi\gamma(\beta\epsilon)^3$ ■

Thus, studying the idealized situation shows why the given algorithm should work.

We also look at the *OptimalTransform()* function. In the case of COMPLETE-MATCH, this is simply the method of Faugeras and Horn and is provably optimal. In the case of PARTIAL-MATCH too, we can still make some claims about its performance. Recall that the proportion of outliers in the data is expected to be less than w; our sample size is r and the size of data set is n. In Algorithm 3, we take $cw^r lg(n)$ samples of size r from the data.

Lemma 2 *In Algorithm 3, the probability that all of the* $cw^r \log(n)$ *samples (each of size* r*) will have at least one outlier i.e. none of the samples is 'good' is less than* $1/n$

Proof: The probability that one sample of size of size r contains an outlier is w^r. Applying Chernoff bounds, [42], we get the required bound ■

We also make the following observation

Lemma 3 *In Algorithm 3, if* d *points agree on a transformation where* $d > \log(1-\alpha)/\log(w)$*, then the probability that all the* d *points are outliers is less than* α*.*

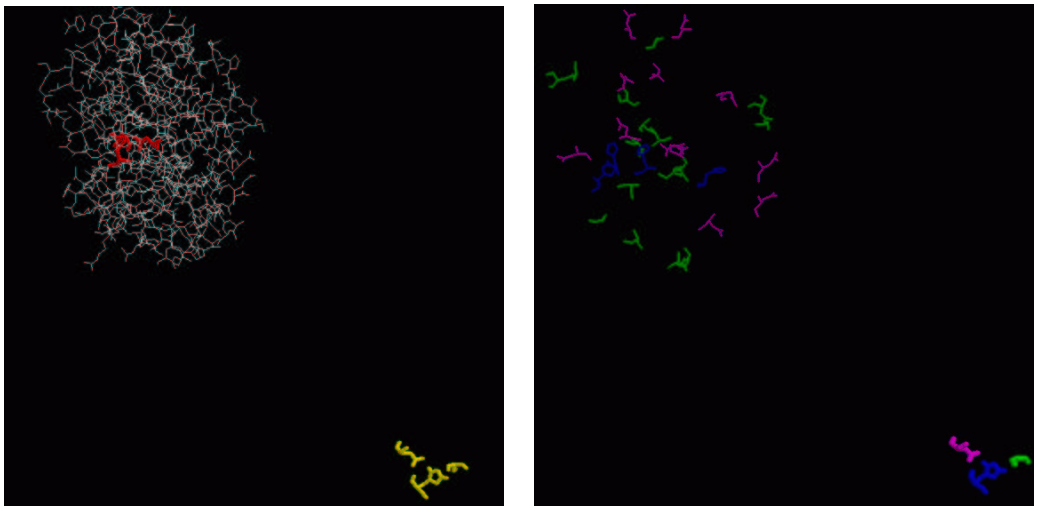
Proof: Applying the inequality $1 - w^d > \alpha$ gives the answer ■

2.6 Results

In this section, we discuss the application of our algorithm to the problem of matching active-sites in proteins.

2.6.1 COMPLETE-MATCH

To give an idea of the extent of pruning (Step 1 in Algorithm 2) done by our algorithm, here are snapshots showing the real size of the protein and the set of possibilities remaining after we had done the pruning.



(a) (b)

Figure 2.9: **Pruning the Search Space**
The above figures show the magnitude of reduction in search space when we use the extra information available to us. The active site to be matched is in the bottom right corner and the target protein is in the top left corner. (a) Before pruning, we will need to search all possible matches. The red colored sub-structure is the actual matching site in the protein (b) After pruning, note how few possible matches are left. Moreover, only amino acids of the same color can be matched (e.g. green with green). Also note, that there are very few blue amino acids in the protein. They can be used for seeding ICP's starting positions.

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 [22]. We chose about 42 different variants of the protease trypsin and about 37 different types of kinases. Choosing the trypsin (consensus) active site [41] as our pattern, we ran our matching algorithm against the trypsins and then against the kinases. Fig 2.9 will help the reader get a feel for the extent of pruning (Step 1 in Algorithm 2) done by our algorithm. The results of our experiments are summarized in the plots shown below (Fig 2.10, 2.11). We were able to detect active sites in the trypsin-like molecules to a high degree of accuracy. Indeed, in 32 of the 42 trypsin-like molecules, we were able to achieve RMSDs less than $0.5\mathring{A}$, indicating a near perfect match. As for the remaining 10 molecules, we looked at the individual structures

to investigate the reason for a higher RMSD. In 3 of these, the X-ray information about the structure was incomplete. In other 4, the sub-structure inside the protein indicated by our algorithm was indeed the active site. However, its structure was somewhat distorted and hence the alignment score was bad. 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. In general, structure based alignment methods will have the problem that they might discover matches which lie inside the surface. When we are looking for active sites, such matches are often irrelevant because active sites are almost always on the molecular surface. This, however, is 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.

2.6.2 PARTIAL-MATCH

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 ($\simeq 8\mathring{A}$) while the other was given a smaller displacement ($\simeq 1.5\mathring{A}$). 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 used the version of the algorithm designed for COMPLETE-MATCH.

We then replaced the the distance measure with one of the M-estimator (we found

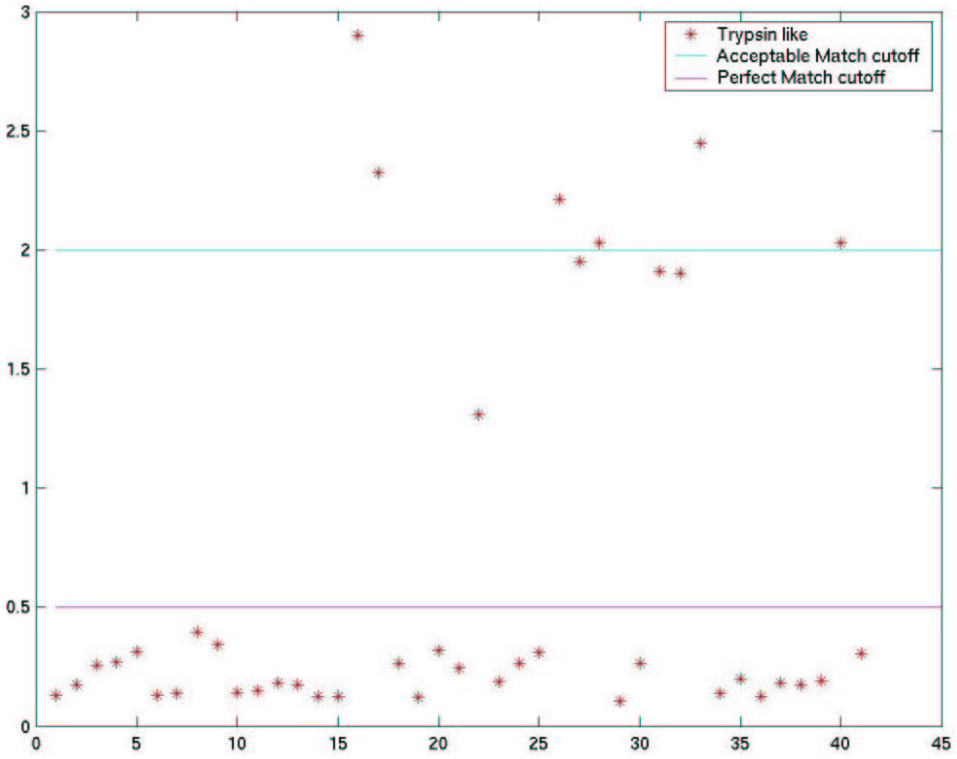


Figure 2.10: **COMPLETE-MATCH on Trypsins**
Y axis: RMSDs between motif (trypsin active site) and (matching part of protein) in $\mathring{A}$. **X axis**: 42 proteins belonging to the Trypsin family. Observe that the quality of the match is near-perfect for most of the proteins. This confirms that all these proteins share the same structural and functional properties.

the Tukey estimator gave the best results) 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()*). As mentioned before, finding such an optimal alignment is hard. In another set of experiments, we used Algorithm 3 to find the best transformation that minimized the error under the distance measure. The following figures (Fig 2.12, 2.13, 2.14) show a typical example of how these three methods behaved. Fig 2.15 summarizes the results of a comparison of these three implementations with a distorted active site.

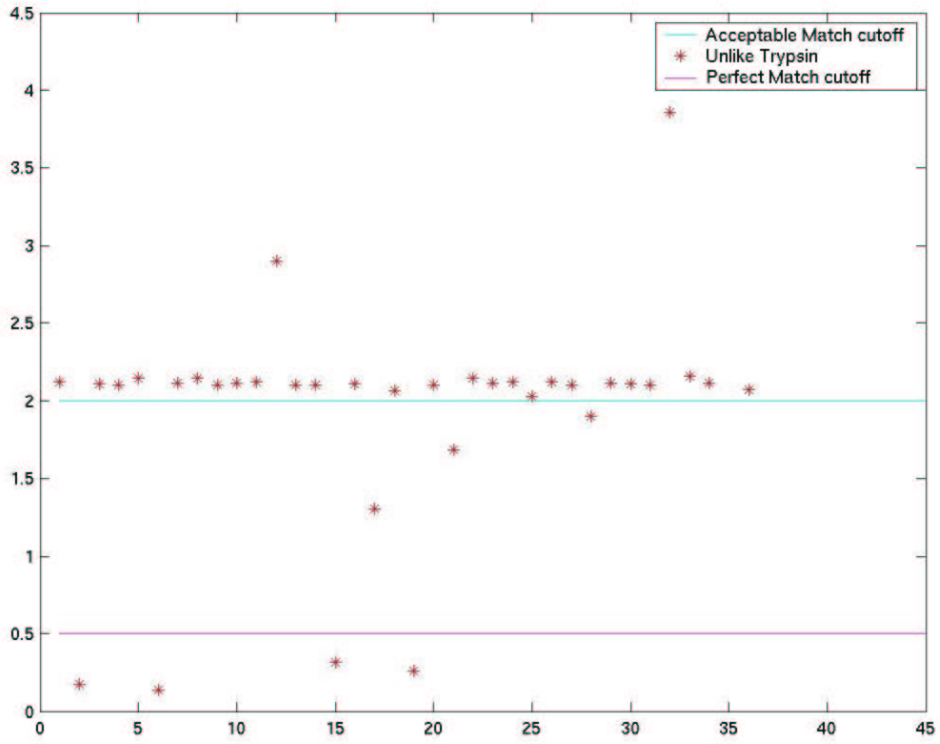


Figure 2.11: **COMPLETE-MATCH on Kinases**
Y axis: RMSDs between motif (trypsin active site) and (matching part of protein) in $\mathring{A}$. **X axis**: 37 proteins belonging to the Kinase family. Observe that most molecules don't have good matches. A RMSD of $2\mathring{A}$ can be achieved because the trypsin active site is small and hence possibilities of chance matches are high. There *are* a few kinase which have very good matches with trypsin active sites. It would be an interesting biological question to explore if these kinases share some properties of trypsins.

2.7 Future Work

Currently, we are exploring the use of our methods for identifying much larger motifs in proteins. There are a lot of open problems in this area. The correct choice of features to match can drastically reduce the search space. Choosing the right ρ for a distance measure in PARTIAL-MATCH requires some more biological understanding of different possible scenarios.

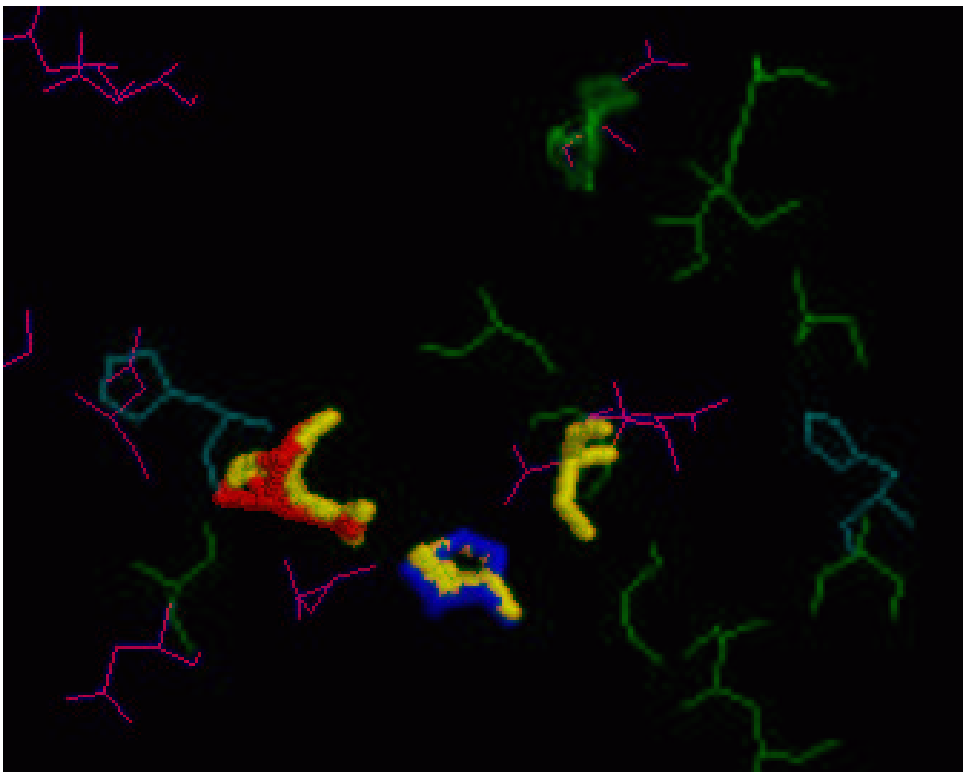


Figure 2.12: **PARTIAL-MATCH**
Solve for COMPLETE-MATCH: $\rho(x) = x^2$: The thick structures are part of the (distorted) active sites we want to match. The thin yellow structures are the part of protein which *should* match. Algorithm 2 (with the distance measure being the same as in COMPLETE-MATCH) tried to find a best match for this out of tune feature but matched it to an incorrect amino-acid. Notice how the thick green structure (upper right quadrant) is mismatched to the wrong amino acid. Because of this, the other two matches (the good ones) also got mixed up

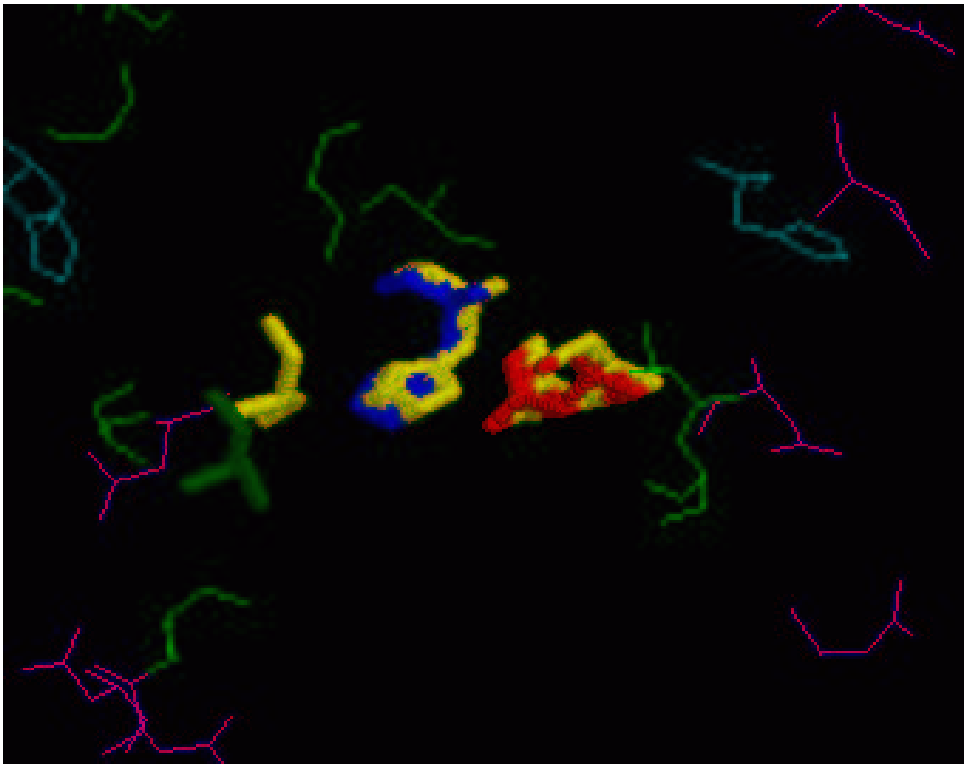


Figure 2.13: **PARTIAL-MATCH**
Tukey Estimator + Generic Optimizer for *OptimalTransform()* : We ran our algorithm with the Tukey estimator (c=2) as the distance function. We did not use Algorithm 3 for finding the optimal transformation. Instead, we used an off-the-shelf package for finding the optimal transformation. This choice performed better than modeling the match as a COMPLETE-MATCH (Fig 2.14) but it was not good enough. Notice how all the thick structures are assigned to the correct yellow corresponding structures, but they are all misaligned. This is because the function to find the optimal transformation had not converged.

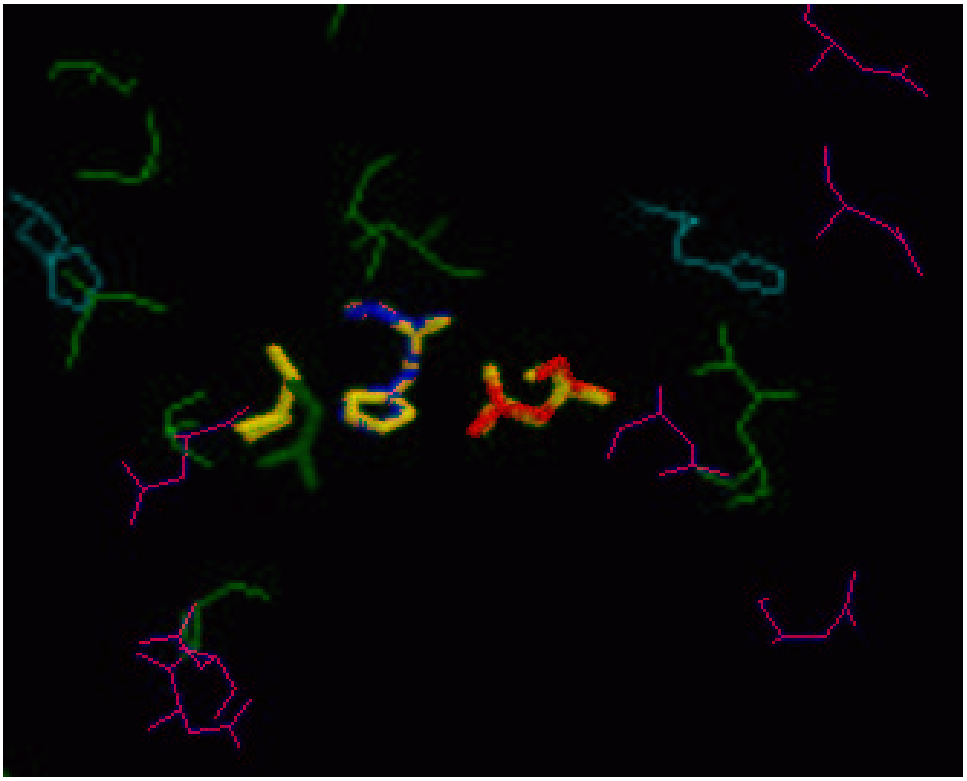


Figure 2.14: **PARTIAL-MATCH**
Tukey Estimator + Algorithm 3 for *OptimalTransform()*: We ran our method (Algorithm 2) with the Tukey Estimator as the distance function and also used Algorithm 3. Here our algorithm chose to ignore the outlier, the dark green feature. Instead, it gave a much better fit on the remaining two features. This is what we wanted

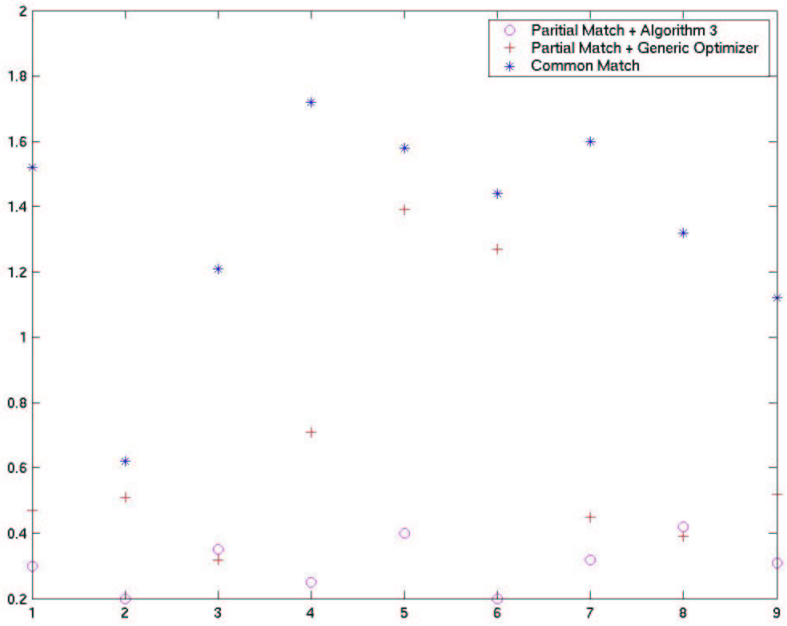


Figure 2.15: **Comparison of partial matching methods**
Y axis: RMSD between the aligned pattern and the matching sub-structure in the protein, *after removing the outlier pair from the match.* The pattern is a distorted version of the trypsin active site. **X axis** 9 different proteins from the Trypsin 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, we can get better performance. If, in addition, we also use Algorithm 3 for aligning the two point-sets, we get the best results.

Chapter 3

Conclusion

In this thesis, the following contributions were made:

Identifying Structural Motifs in Proteins We have presented a rigorous formulation of the problem of finding a match for a given substructure in a protein. Unlike previous approaches, our formulation can provide quantitative scores for even partial matches. We then introduce some robust matching and estimation techniques from other fields in computer science and statistics and show they can be adapted to the purpose of matching protein structures. We then discuss ways to enhance these algorithms by pruning the search of best matches by using the extra information available with proteins.

In addition to these, I have also worked on some other problems. In particular, I have been involved in designing and evaluating an algorithm that enables us to (approximately) solve the inverse kinematics problem for protein chains in an efficient and incremental manner. The algorithm depends on precomputation of solutions for short chains of proteins and uses the information to make quick, incremental modifications to the protein chain. Along with M Serkan Apaydin, I have also worked on identifying and solving the problems in applying Stochastic Roadmap Simulations to proteins of significant size and complexity.

Bibliography

[1] *http://www.expasy.ch/sprot*

[2] I. Lotan, F. Schwarzer, D. Halperin and J.C. Latombe *Efficient Maintenance and Self-Collision Testing for Kinematic Chains* To appear in ACM Symposium on Computational Geometry, 2002

[3] M.S. Apaydin, D.L. Bruglag, C. Guestrin, D. Hsu, and J.-C. Latombe *Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion* International Conference on Research in Computational Biology (RECOMB), 2002

[4] F. Schwarzer and I. Lotan *Efficient Nearest Neighbour Search in Protein Conformations* in preparation for publication.

[5] V.S. Pande, A.Y. Grosberg, T. Tanaka, and D.S. Rokhsar *Pathways for protein folding: Is a new view needed?* Current Opinion in Structual Biology, 8:68–79, 1998

[6] M. Apaydin, C. Guestrin, Ch. Varma, D. Brutlag *Stochastic Roadmap Simulation for The Study of Ligand-Protein Interactions* European Conference on Computational Biology

[7] Bojan Zagrovic, Eric Sorin, and Vijay Pande *Beta Hairpin Folding Simulations in Atomistic Detail* Journal of Molecular Biology, 2001

[8] L.M. Kauvar, H.O. Villar *Deciphering Cryptic Similarities in Protein Structure*, Current Opinions in Biotechnology 9:390-394, 1998.

[9] 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.

[10] 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.

[11] C Vita, J Vizzavona, E Drakopoulou, S Zinn-Justin, B Gilquin, A. Menez, *Novel miniproteins engineered by the transfer of active sites to small natural scaffolds* Biopolymers 47(1):93-100, 1998

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

[13] 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

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

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

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

[16] 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

[17] J. Foley, A. van Dam, S. Feiner and J. Hughes *Computer Graphics: Principles and Practice* Addison-Wesley, Reading, MA, 1995

[18] I.J. Dryden, *General shape and registration analysis*, In W. S. Kendall, O. BarndorffNielsen, and M. N. M. van Lieshout (Eds.), SEMSTAT 3, London. Chapman and Hall, 1997.

[19] P.J. Besl, *Geometric Modeling and Computer Vision*, Proc. IEEE, Vol. 76, pp. 936-958, 1988

[20] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, *Basic local alignment search tool*, J. Mol. Biol. 215:403-410, 1990

[21] M. Levitt, M. Gerstein *A unified statistical framework for sequence comparison and structure comparison* Proc Natl Acad Sci U S A. May 26;95(11):5913-20, 1998

[22] Brookhaven Protein Data Bank *http://www.rcsb.org/pdb/*

[23] Rasmol: a program for visualizing molecular structure *http://www.umass.edu/microbio/rasmol/*

[24] SwissPDB Viewer: a program for visualization and manipulation of molecular structure *http://www.expasy.ch/spdbv/*

[25] M.L. Connolly *Analytical molecular surface calculation*, J. Appl. Crystallogr. 16, 548-558, 1983

[26] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry, Algorithms and Applications* Springer Verlag, 1997

[27] M.H. DeGroot and M.J. Schervish *Probability and Statistics*, 3^{rd} edition. Addison Wesley, 2002

[28] 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.

[29] Z. Zhang *Iterative point matching for registration of free-form curves and surfaces*, International Journal of Computer Vision, Vol 13(2) pp. 119-152, Oct 1994

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

[31] Y. Lamdan and H.J. Wolfson, *Geometric Hashing: A General and Efficient Model-Based Recognition Scheme*, In Proceedings of the IEEE Int. Conf. on Computer Vision, pages 238-249, Tampa, Florida, December 1988.

[32] 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

[33] X. Pennec, N. Ayache *A geometric algorithm to find small but highly similar 3D substructures in proteins*, Bioinformatics (ex Cabios), 14(6), pages 516-522, July 1998

[34] G. Barequet, M. Sharir *Partial surface matching by using directed footprints*, Comput. Geom. Theory Appls. 12, 45-62, 1999

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

[36] 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

[37] Z. Zhang, *Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting* International Journal of Image and Vision Computing, Vol.15, No.1, pages 59-76, January 1997

[38] P.H.S. Torr, *Outlier Detection and Motion Segmentation* PhD thesis, Dept. of Engineering Science, University of Oxford, 1995

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

[40] P. Koehl *Personal Communications*

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

[42] R. Motwani and P. Raghavan *Randomized Algorithms* Cambridge University Press, 1995

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

[44] S.C. Lovell, J.M. Word, J.S. Richardson, D.C. Richardson, *The Penultimate Rotamer Library* Proteins 15;40(3):389-408, Aug 2000

[45] R.L. Dunbrack Jr, M. Karplus, *Conformational Analysis of the Backbone-dependent Rotamer Preferences of Protein Sidechains* Nature Struct Biology 1(5):334-40 May 1994

[46] P. Koehl, M. Delarue, *Application of a Self-consistent Mean Field Theory to Predict Protein Side-chains Conformation and Estimate their Conformational Entropy* J Mol Bioln 3;239(2):249-75 Jun 1994