M.Tech. (Computer Science) Dissertation Series

Finding 3D Structure of Proteins using Characteristics of Short Sequences

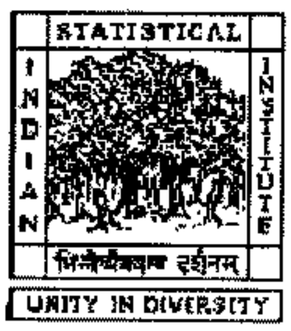
a dissertation submitted in partial fulfillment of the
requirements for the M.Tech. (Computer Science)
Degree of the Indian Statistical Institute

By

Sudeepta Kumar Ojha

under the supervision of

Prof. Nikhil Ranjan Pal
ECSU



INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road
Calcutta-700035

Certificate

This to certify that **Mr. Sudeepta Kumar Ojha**, student of Final year Computer Science, Master of Technology, Indian Statistical Institute, Kolkata, has successfully completed his dissertation work titled "***Finding 3D Structure of Proteins using Characteristics of Short Sequences***" under my supervision. To the best of my knowledge this work has not been submitted elsewhere for the award of any degree or research publication.

Prof. Nikhil Ranjan Pal
ECSU, ISI Kolkata

Date: 13·07·2005

Acknowledgement

Only sincere motivation and proper guidance can lead to the accomplishment of any research work, besides the personal efforts.

At the onset, I would like to express my sincere gratitude to my dissertation supervisor **Prof. Nikhil Ranjan Pal**, ECSU, Indian Statistical Institute, Kolkata, for giving me a valuable opportunity to do this dissertation work under his supervision.

I am highly indebted to him, for his immense cooperation, valuable suggestions, inspirational guidance and critical remarks, provided with the personal touch without which it would have been next to impossible for me to complete the project successfully.

I would also extend my sincere gratitude to all my teachers for their direct and indirect cooperation and support. I would also like to extend my special thanks to Mr. Somitra Kr. Sanadhya, Research Fellow, ECSU, Indian Statistical Institute, Kolkata, for his valuable help and all Staff of ECSU for their continuous support.

My final thanks go to my friends who helped me a lot through their valuable suggestions in completing this report.

List of Figures

List of Tables

Contents

1 Introduction to Problem

1.1 Introduction

Proteins are the most structurally complex macromolecules known. They are long chain of molecules. They can be regarded as necklaces of 20 different amino acids that are arranged in different order to make chains of up to thousands of amino acids. The result is an extreme variety of proteins, each type with its own unique structure and function. In order to carry out their function, each protein must take a particular shape, known as its *fold*. When a protein is put into a solvent, within a very short time it takes a particular 3D shape. This self assembling process is called *folding*.

Sometimes the proteins *misfold* (i.e. do not fold correctly) and they can aggregate. Aggregation of misfolded proteins is believed to be the cause of some disorders such as Alzheimer's diseases, Parkinson's disease, prion disease (e.g., "mad cow" disease) and some cancers. The diverse range of diseases that results from protein misfolding has made this a subject of intense investigation: learning how proteins fold will teach us how to design protein-sized "nano-machines" that can do similar tasks and it will help us to prevent or reverse diseases in which proteins have departed from the correct folding route. However, it is very time consuming to find the 3D structure of a protein using X-ray Crystallography or Nuclear Magnetic Resonance(NMR) imaging. Hence, researchers are working on finding computational methods for protein fold prediction. In this thesis we shall propose some methods to predict 3D structures of proteins from its amino acid sequences exploiting statistical information available in proteins with known 3D structures. In particular we made the following contributions.

1. We proposed a mechanism for generation of self-organizing map for structures called *Structural Self-Organizing Map(SSOM)*. This method can be applied in areas other than protein folding also.

2. We proposed a modified form of mountain clustering called *Structural Mountain Clustering Method(SMCM)* that is very effective for the problem understudy and is simpler.

3. The Structural Self-Organizing Map is then augmented by two subclustering methods resulting in two schemes for building block generation.

4. We applied these three new methods to find representative hexamers from a given data base and compared performance of the proposed schemes to an existing method.

5. We then used the extracted hexamers to reconstruct some proteins. The results are quite good.

1.2 Proteins

Proteins form the very basis of life. They regulate a variety of activities in all known organisms, from replication of the genetic code to transporting oxygen, and are generally responsible for regulating the cellular machinery and consequently, the phenotype of an organism. Proteins accomplish their task by three-dimensional tertiary and quaternary interactions between various substrates such as DNA and RNA, and other proteins. Thus knowing the structure of a protein is a prerequisite to gain a thorough understanding of the proteins' function.

Proteins make up about 15% of the mass of an average person. They carry out vital functions in every cell and are essential to us in an enormous variety of different ways. Muscles, cartilage, ligaments, skin and hair are mainly protein materials. Proteins play also a vital role in keeping our body working properly. Hemoglobin (that carries oxygen to our tissues), hormones (such as insulin, that signals our bodies to store excess sugar), antibodies and enzymes are all examples of proteins. Amino acids are the molecular units that make up proteins; they are organic compounds, formed from carbon, hydrogen, nitrogen, oxygen, and sulfur (with sulfur only present in R, the so called residue or side chain). All amino acids have the same general formula shown in Figure 1.

Proteins are macromolecules constructed from one or more chains of amino acids (i.e. they are polymers) that encode their 3D structure. To form a protein, the amino acids are linked by peptide bonds; that is why the chain of amino acids (protein) is known as a polypeptide (see Figure 1). To synthesize proteins, "machines" called ribosomes string together amino acids into long, linear chains; this process is called translation and only 20 amino acids are normally used. These chains loop about each other in a variety of ways (they fold), but only one of these many ways allows the protein to function properly. In other words, when a specific protein is synthesized in a living system, that protein rapidly assumes a configuration specific for its type and its function depends primarily on its configuration rather than on its specific amino acid sequence. A typical protein contains 200-300 [2] amino acids but some are much smaller and some much bigger (the largest is *titin*, a protein found in cardiac muscle: it contains more than 26920 amino acids in a single chain).

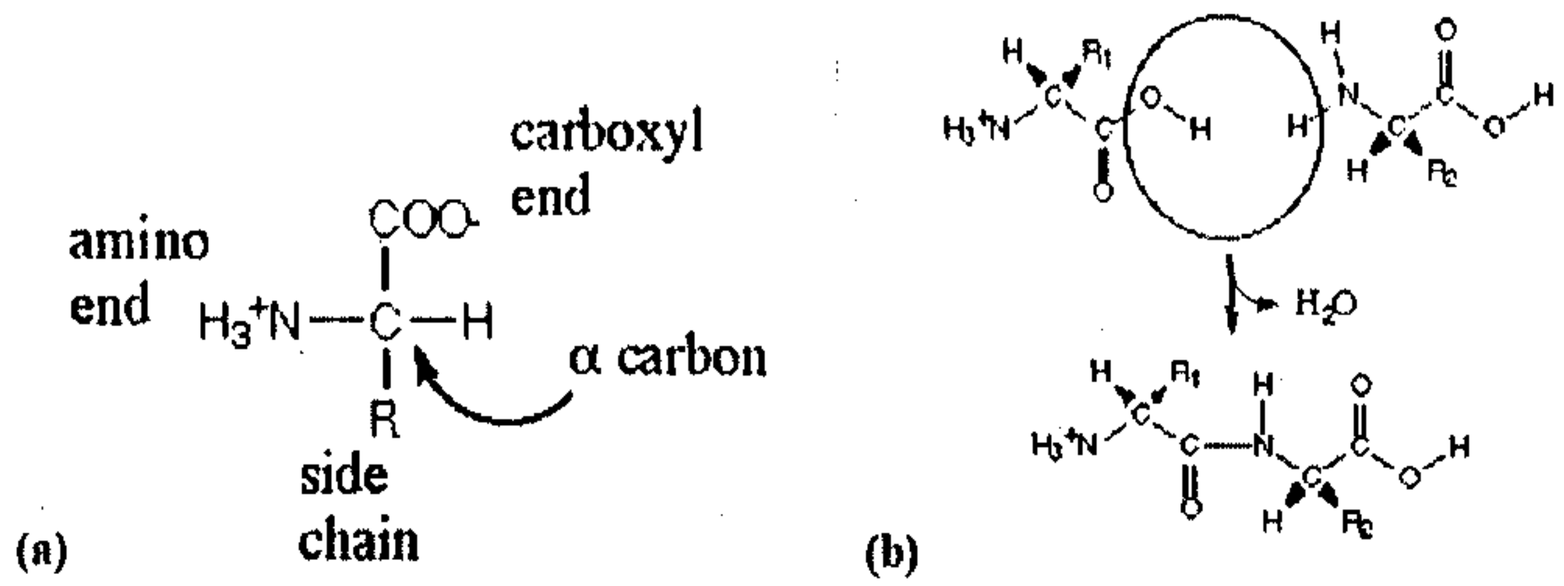


Figure 1: (a) General formula of amino acids. R is called residue or side chain and it is made of various combinations of C,H,N,O and S. (b) Peptide bonds link amino acids. They are formed via a dehydration synthesis reaction between the carboxyl group of the first amino acid with the amino group of the second amino acid.

More than half a century ago, Linus Pauling (Nobel prize, 1954) discovered that a major part of most proteins' folded structure consists of two regular, highly periodic arrangements of amino acids, designated *"a"* and *"b"*. The key to both structures is the hydrogen bond, that stabilizes the structures. The *"a"* structure is now called α-helix (Figure 2). It is a spiral configuration of a polypeptide chain stabilized by hydrogen bonds between the CO group of one amino acid at position n and the NH group of the amino acid which is four residues away (n+4).

The *"b"* structure is now called β-sheet (Figure 3). It is an essentially flat 2D structure of parallel or anti-parallel β strands; each β strand consists of two polypeptide chains that are (almost) fully extended and hydrogen-bonded to each other. All other local arrangements that are neither α-helix nor β-sheet are described as random coil: they are random in the sense that they are not periodic.

Proteins have multiple levels of structure:

1. Primary structure: Linear structure determined solely by the number, sequence, and type of amino acid residues (R).

2. Secondary structure: Local structure determined by hydrogen bonding between amino acids and non-polar interactions between hydrophobic

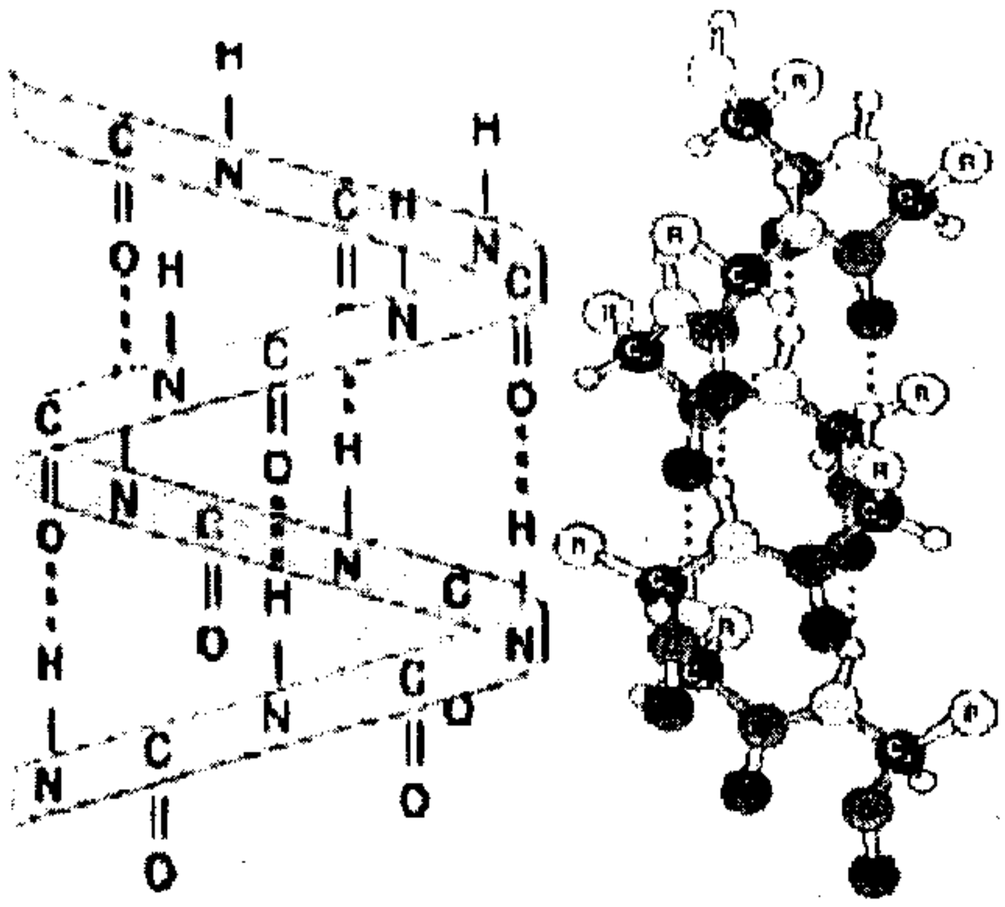


Figure 2: Spiral configuration of the α-helix structure. Hydrogen bonds between the CO group of one amino acids and the NH group of another amino acid hold α-helices together.

regions. These interactions produce, in general, three secondary structures: helix(Figure 2), β sheet (Figure 3), and random coil.

3. Tertiary structure: It results from various interactions (mainly hydrophobic attractions, hydrogen bonding, and disulfide bonding) of the amino acids side chains (R) that pack together the elements of the secondary structure. The result is a 3D configuration of proteins (Figure 4).

4. Quaternary structure: It is characterized by the interaction of two or more individual polypeptides (often via disulfide bonds) and the result is a larger functional molecule (hemoglobin, Figure 4)

1.3 The protein folding problem

In the early 1960s, Anfinsen [3], [4] showed that proteins actually tie themselves: if proteins become unfolded, they fold back into proper shape, no shaper or folder is needed. It is a self-assembling process. Sometimes a protein will fold into a wrong shape. Partially folded chains can exist but don't

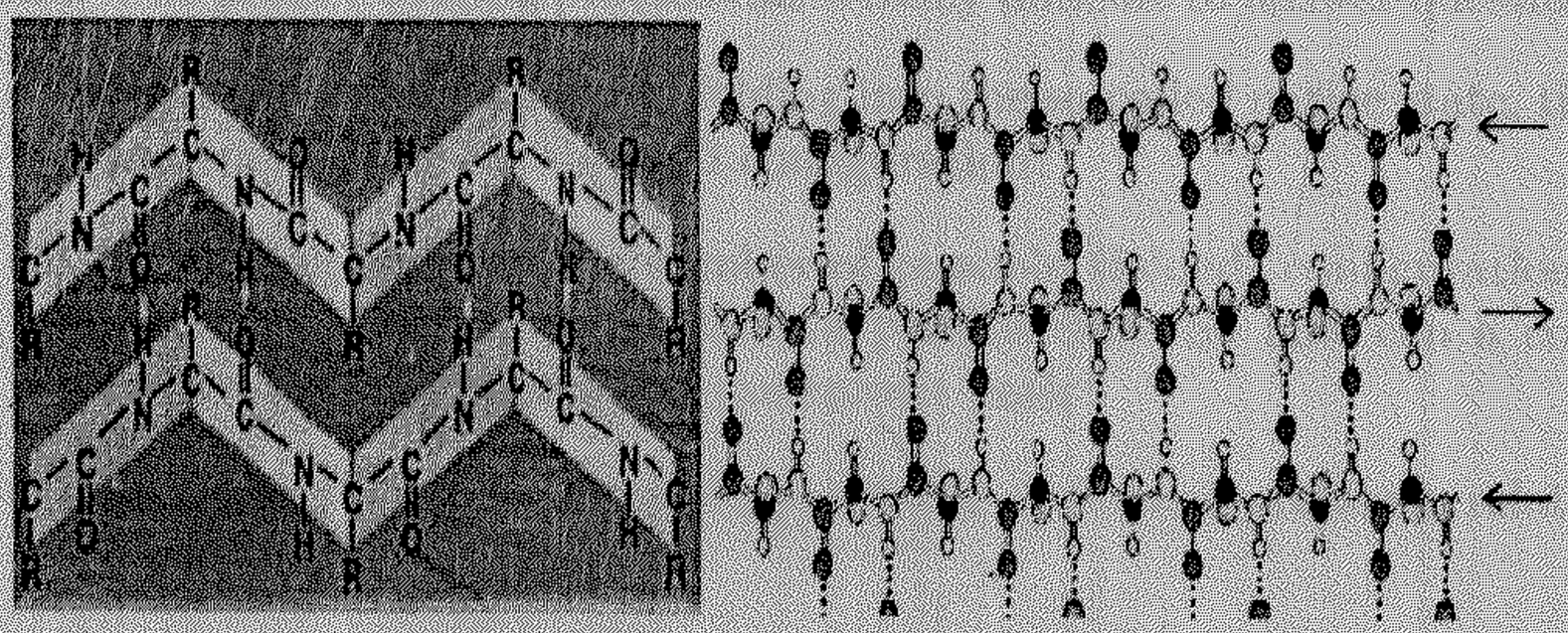


Figure 3: β-pleated sheets structure is stabilized by hydrogen bonds between nitrogen atoms (of the NH group of one amino acid) and oxygen atoms (of the CO group of another amino acid) of two adjacent chains.

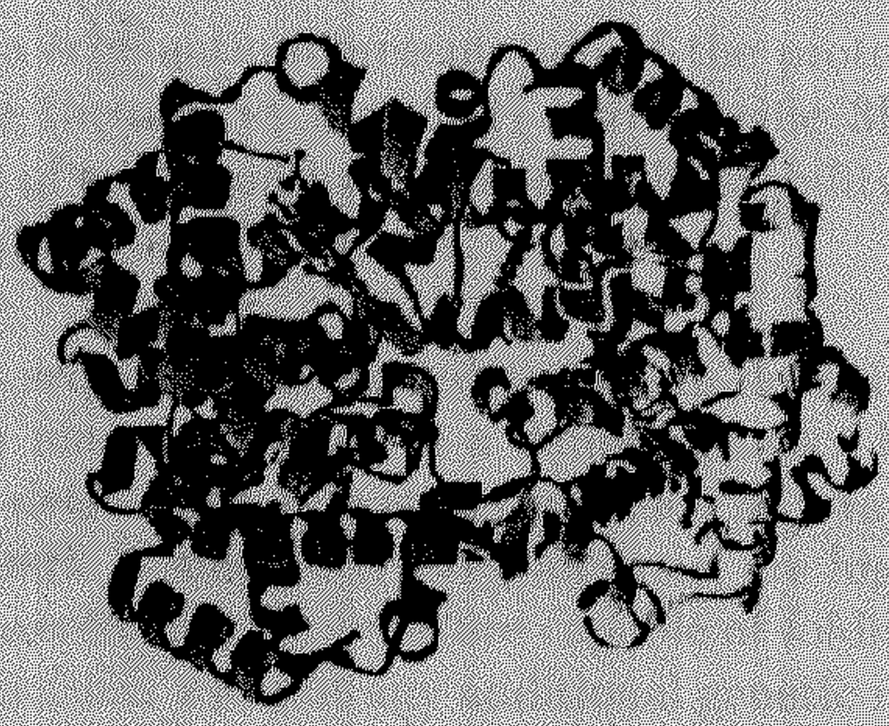


Figure 4: Many proteins are combination of several proteins. They are aggregates of smaller globular proteins, most frequently identical subunits of the same protein. In the picture the four different subunits of the hemoglobin are shown in four different shades.

stay that way very long; they become fully folded chains in a fraction of a second. The protein folding process raises many difficult to answer questions.

- What rules govern the rapid folding into the so-called native state, that is the energetically stable, 3D configuration?
- Given a polypeptide, the number of possible (folded) configuration is enormous: hundreds of millions of possibilities. So how can a completely unfolded protein find the correct path?
- How many different folding routes exist, and what are their relative probabilities?
- If a particular protein always assumes the same configuration in a living system (its "native configuration"), and if that configuration represents some sort of energy minimum for the polypeptide chain, how does the protein find that energy minimum within nanoseconds?
- Does the protein pass through every possible configuration state until the energy minimum configuration is discovered?
- Is there a critical intermediate partially folded configuration? Are there constraints that reduce the number of possible configurations?

Figure 5 depicts a schematic representation of the protein folding problem. Proteins organize themselves (i.e. fold) into specific 3D native structures through a myriad of conformational changes, the stability of which is defined by innumerable forces between atoms.

Once a protein sequence has been determined, deducing its unique three-dimensional (3D) native structure is a daunting task. Experimental methods to determine detailed protein structure, such as x-ray diffraction studies and nuclear magnetic resonance (NMR) analysis, are highly labor intensive. Since it was discovered that proteins are capable of folding into their unique functional 3D structures without any additional genetic mechanisms, over 25 years of effort has been expended into the prediction of 3D structure from sequence. Despite the large amount of effort expended, the protein folding or protein structure prediction problem, as it has come to be known, remains largely unsolved.

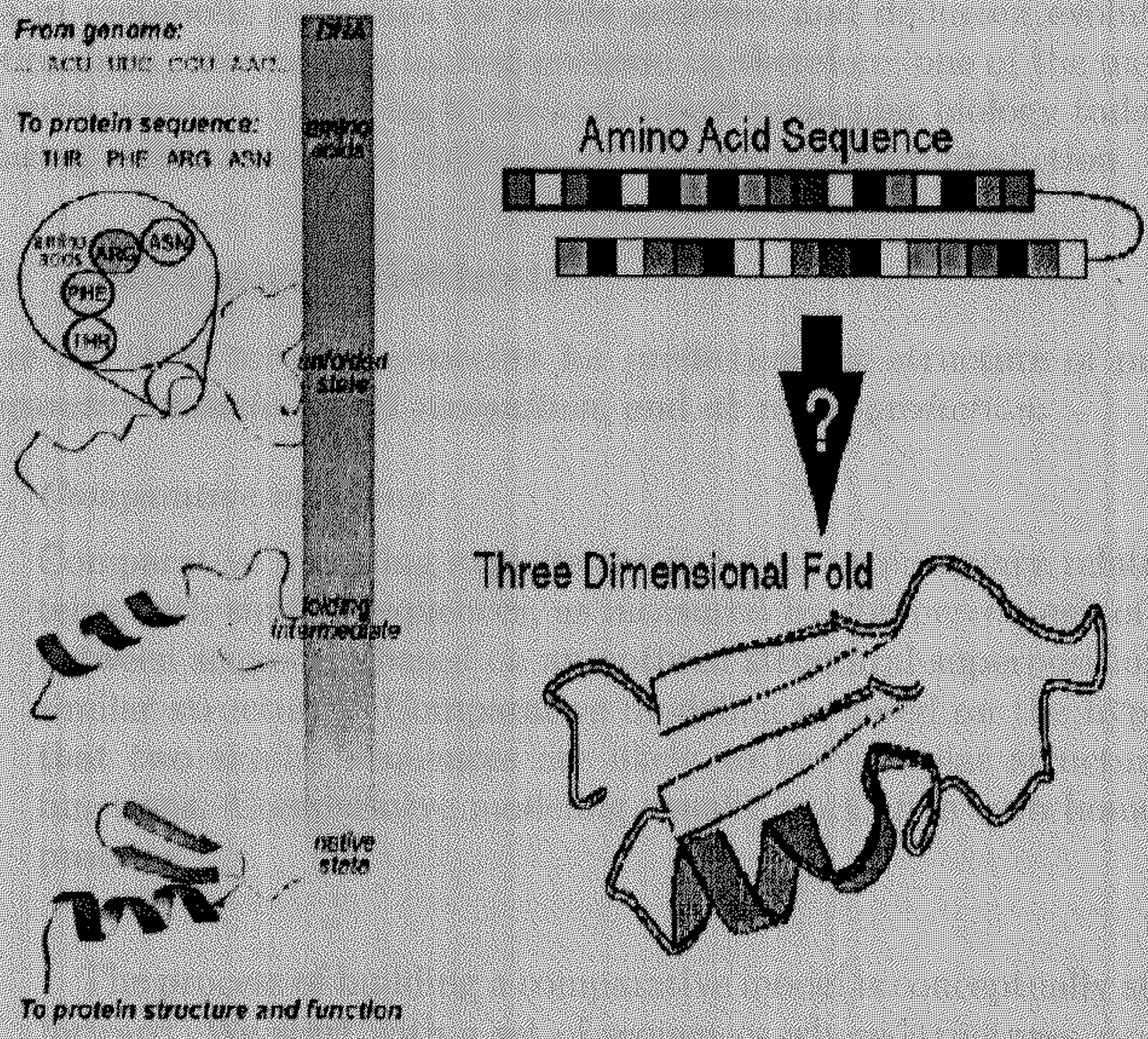


Figure 5: Schematic representation of the protein folding problem. Proteins organize themselves (i.e. fold) into specific 3-D native structures through a myriad of conformational changes, the stability of which is defined by innumerable forces between atoms.

Knowing the structure of a protein sequence enables us to probe the function of the protein, understand substrate and ligand binding, devise intelligent mutagenesis and biochemical protein engineering experiments that improve specificity and stability, perform rational drug design, and design novel proteins. Understanding structure has potential applications in the various genome projects being undertaken, such as mapping the functions of proteins in metabolic pathways for whole genomes and deducing evolutionary relationships. The protein folding problem is, therefore, one of the most fundamental unsolved problems in computational molecular biology today. The following section describes methods for protein structure prediction.

1.4 Theoretical Methods

There are three major approaches for predicting the structure of proteins: *comparative modeling*, *fold recognition*, and *ab initio prediction*.

1. *Comparative modeling*

   Comparative modeling exploits the fact that evolutionary related proteins with similar sequences, as measured by the percentage of match between sequences based on an optimal structural superposition, have similar structures. Comparative modeling is based on the hypothesis that two similar sequences are likely to have similar structures and functions. Comparative modeling usually works fine with homologous proteins.

   The process of building a comparative model is conceptually straightforward. We start with a set of sequences whose structures are known (determined by experimental methods). This set is called *training set*. Then an alignment is performed between each sequence in the *training set* and with the sequence to be modeled (the target). The fold of the best matched sequence can be assigned to the target.

   Instead of computing explicit match between sequences; the inherent characteristics of the *training set* can be implicitly modeled using computational systems like neural networks [5]. The trained neural networks can then be used to predict the folds of new protein sequences. In this method we predict the fold, but do not find the actual 3D structures.

2. *Fold recognition or "Threading"*

Threading uses a database of known three-dimensional structures to match sequences without known structure. This is accomplished by the aid of a scoring function that assesses the fit of a sequence to a given fold. These functions are usually derived from a database of known structures and generally include a pairwise atom contact and solvation terms. Threading methods compare a target sequence against a library of structural templates, producing a list of scores. The scores are then ranked and the fold with the best score is assumed to be the one adopted by the sequence [6]. The methods to fit a sequence against a library of folds can be extremely expensive computationally. Such threading processes may involve dynamic programming, Gibbs Sampling using a database of "threading" cores, Simulated annealing, Monte Carlo method and branch and bound heuristics, or as "simple" as using sophisticated sequence alignment methods such as Hidden Markov Models.

3. *Ab initio prediction*

   The ab initio approach is a mixture of science and engineering. The science is in understanding how the three-dimensional structure of a protein is attained. The engineering portion is in finding the three-dimensional structure from a given the sequence. The ab initio folding process can be broken down into two components: devising a scoring function that can distinguish between correct/good (native or native-like) structures from incorrect (non-native) ones, and a search method to explore the conformational space. In many ab initio methods, the two components are coupled together such that a search function drives, and is driven by, the scoring function to find native-like structures.

Currently there does not exist a reliable and general scoring function that can always drive a search to a native fold, and there is no reliable and general search method that can sample the conformation space adequately to guarantee a significant fraction of near-natives (< 3.0 Angstroms RMSD from the experimental structure).

Some methods for ab initio prediction include Molecular Dynamics (MD) simulations of proteins [7]; Monte Carlo (MC) [8] simulations that do not use forces but rather compare energies, via the use of Boltzmann probabilities; Genetic Algorithms [9] which tries to improve on the sampling and the convergence of MC approaches, and exhaustive and semi-exhaustive lattice-based studies [10] which are based on using a crude/approximate fold representation (such as two residues per lattice point) and then exploring all or large amounts of conformational space given the crude representation.

1.5 What can structure prediction do for us?

Given the large volume of genes being sequenced, the rate of discovery of new protein sequences is growing exponentially relative to the rate with which protein structures are being solved by experimental methods. In many situations, even a crude or approximate model can help an experimentalist significantly in guiding his/her experiments. Thus even though the current methods are still in their infancy, prediction of structures for all protein sequences of complete genomes in conjunction with experimental work is a realistic goal. Structural analysis of proteins is in great demand for further mutagenesis, substrate and inhibitor design, and for enhancing function and stability. Methods such as molecular dynamics simulations use structural data and methods for structure prediction to probe protein and organizational function and evolution. It can help in predicting functions of a protein. It may also help in designing patient specific genetic drugs for different difficult disorders.

2 Approaches to Prediction of Proteins Structure

Understanding the relationship between the three-dimensional structure of proteins and their one-dimensional amino acid sequence is still one of the most fundamental unsolved problems in physical biochemistry. Much attention has been given [11], [12], [13], [14] to the relationship between structure and sequence of short oligopeptides.The idea is to see to what extent identical or similar sequences imply similar structures.

In [1] an algorithm has been proposed for analyzing and ultimately predicting protein structure, defined at the level of C_α coordinates. Our investigation is primarily motivated by the method described in [1]. We analyze hexamers (oligopeptides of six amino acids residues) and show that their structures tends to concentrate in specific clusters rather that vary continuously. Thus, using a limited set of standard structural building blocks taken from these clusters as representatives of the repertoire of observed hexamers, one may be able to construct 3D structures of proteins. After replacing each hexamer by a standard building block with similar conformation, approximate reconstruction of the actual structure can be done by smoothly joining the overlapping building blocks into a full protein.

Specifically, we re-address the three following questions in [1]:

1. Is it possible to divide a given set of hexamers obtained from a set of proteins into a reasonable number of really different structural shapes? Whether the conformations of hexamers vary continuously or can be separated into disjoint clusters.

2. Can the library of these different shapes, which are called building blocks, be used to reconstruct the structure of proteins?

3. Do the building blocks carry some sequence specificity that will enable us to assign building blocks from sequences, and thus use them in a three dimensional prediction scheme?

In this thesis our objective is to develop mechanisms for finding proto-typical hexamers from a set of proteins whose 3D structures are known. These prototypes will then be used to construct the 3D structure of proteins with unknown structures. In this regard, we propose a new Self-organizing map algorithm for structures and shapes. We call it *Structural Self-organizing(SSOM).*

Table 1: Refined Brookhaven Peptides.

lAPR	1BP2	1CC5	lCCR	lCPP	lCPV	lCRN	lCTF
lECA	1FB4h	lFBJl	1FC2d	lFDX	lGAPa	lGCR	lHIP
lHMQa	lINSa	2INSb	lLHl	lLZl	lLZT	lMBD	lNXB
lPCY	lPP2r	1PPD	lPFT	lSBT	1SN3	lTGSi	2ABXa
2ACT	2ALP	2APP	2AZAa	2CAB	2CCYa	2CDV	2CTS
2CYP	2ESTe	2FD1	2GN5	2INSa	2LHB	2LZM	20vo
2PABa	2PKAa	2PKAb	2RHE	2SGA	2SNS	2SODo	351C
3C2C	3DFR	3ICB	3PGM	3PTP	3RP2a	3RXN	3SGBe
3TLN	4ADH	4APE	4ATCa	4ATCb	4CYTr	4DFRa	4FXN
4HHBb	4HHBc	4HHBd	4SBVa	5CPA	5LDH	5PTI	5RSA
5RXN	7CAT						

We also propose a modified version of Mountain Clustering Method called *Structural Mountain Clustering Method*(SMCM). We compare performance of our methods with methods in [1]. Before describing the methodology we first describe the data set used.

2.1 Database

Our structural data base was taken from the Brookhaven Protein Data Bank, as released in January 15, 1987 (354 polypeptide chains, 61,064 residues). This data set was used by the authors of [1]. The retained structures are only those structures for which X-ray data had been collected to 3.0 A or higher resolution, and which had been refined against the observed X-ray data to an R factor of less than 30%. Finally we use a library of 82 peptides (12,973 residues), which are referred to as the refined Brookhaven data base (see Table 1) Actually, all 82 structures had been solved to a resolution of 2.8 A or higher of which 68 structures had a resolution of 2 A or better.

2.2 Defining the distance between protein structure

The distance (or similarity) between two structures is not easy to define. We used the following well-accepted definition. The RMS deviation distance between two structures **s** and **t** is measured by first aligning them to the greatest possible extent using the BMF (best molecular fit) algorithm of Nyburg [15] or Kabsch, [16], [17] and then calculating the difference in the positions of the corresponding C_{α} atoms as a normalized root mean square

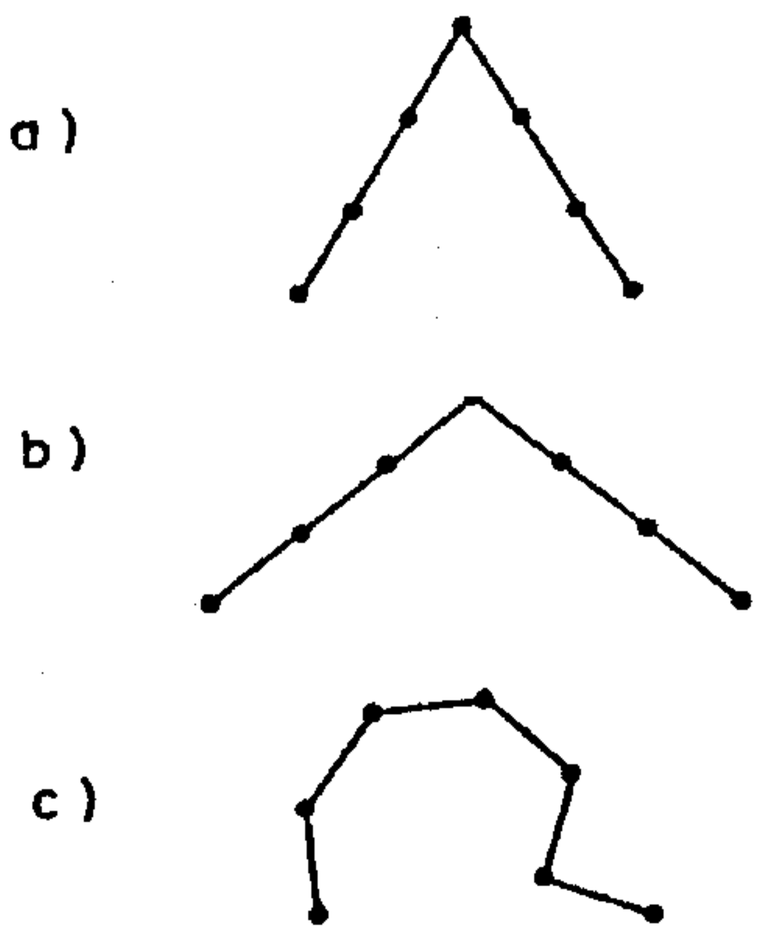


Figure 6: The RMS distance between (a) and (b) is larger than the distance between (a) and (c). But, the shape of (a) is more similar to (b) that to (c).

deviation.

$$RMS = \left[\frac{\sum_{i=1}^{n} \|\mathbf{r}_i^s - \mathbf{r}_i^t\|^2}{n-2}\right]^{\frac{1}{2}}, \tag{1}$$

where $\mathbf{r}_i^s$ is 3D coordinate of i^{th} atom in molecule s and n denotes the number of atoms in molecule.
One must keep in mind that this definition does not always capture the intuitive notion of similarity.

- First, it is highly scale dependent, i.e., two structures with a similar shape but different sizes will show no similarity.
- Second, it is not sensitive to the geometrical and topological properties of the structures, see for example Figure 6.
- Third, and possibly the most important, it is sensitive to insertions and deletions since it is based on the distance between corresponding atoms in the two structures.

However, for short structures such as hexamers, the RMS distance seems to be a good measure of similarity.

2.3 Measuring the random distance between protein structures

In order to evaluate the statistical significance of our results, we need a framework of measurements of random distances between protein fragments of various lengths. We defined random distance [1] as the expectation of the RMS distance between a pair of fragments, of given length, drawn at random from our refined library. This distance is calculated by choosing a few sample proteins of different types, extracting all of their overlapping fragments, and calculating the average distance between them. In this study, we used four proteins: 4HHBb (human deoxyhemoglobin, p-chain), 5PTI (bovine pancreatic trypsin inhibitor), 1BP2 (bovine pancreatic phospholipase A2), and 1PCY (oxidized poplar plastocyanin), with total length of 426 residues. These structures had been very accurately determined, and they represent different structural classes of proteins.

2.4 Methods for selection of building blocks

Throughout this analysis hexamers(fragments of length 6) are used, which appear to be long enough to carry structural meaning. The detailed reasons for the selection of this length will be elaborated in the discussion.

- Two Stage Clustering Algorithm [1]
  We describe the algorithm in [1] as we shall compare our method with this one. In this approach, a cluster is defined as a set of structures with the property that the RMS deviation between members, or alternatively, from some typical member, is less than some fixed value. Since 1 Å [1] seems to be the separation point between similar and not similar hexamers, 1 Å was selected as the threshold value for the clustering process.

The same four sample proteins were used. Each protein was divided into overlapping hexamers; thus, for a protein of length N there were N-5 hexamers, and for these four proteins a total of 406 hexamers. The RMS distance between each of the 82,215 pairs of hexamers was calculated (this number is simply [K(K-1)1/2] for K=406). The clustering procedure consisted of two stages. In the first stage, a variant of the K-nearest neighbor clustering algorithm is used. A hexamer is selected to be the first member in the first cluster, and all other hexamers closer to this first member than the 1 Å threshold value are assigned to the same cluster.. Each member of the cluster then serves as a new source

to add all of its sufficiently close neighbors to the cluster. This "annexation" process is repeated until no further hexamers can be added to the cluster. A new hexamer is then selected as the first member of the next cluster, which is constructed in the same way. The procedure is terminated when all the hexamers have been assigned to clusters. This part of the algorithm is deterministic, i.e., the assignment to clusters is independent of the order in which the hexamers are used, and it has a run-time complexity that is quadratic in the number of elements to be clustered.

In the process described above, the diameter of a cluster (i.e., the greatest distance between any pair of members) can grow significantly. Consequently, in the second stage of the procedure a second algorithm to obtain a finer subclustering assignment is applied. For dividing each cluster into subclusters, each of which contains a member whose distance from any other member is not more than a threshold value, and again 1 Å is used. This member is called the center of the cluster. Optimal clustering, in the sense that the number of subclusters is minimal, is computationally very expensive so a heuristic procedure is applied: For each cluster, the member having the maximum number of neighbors is chosen as the center of a new subcluster containing those neighbors as members. The process is repeated for the other unassigned hexamers until all the hexamers of the cluster are assigned to subclusters.

- Structural Mountain Clustering Algorithm We now propose a modified form of a clustering method due to Yager [18].
  Mountain Clustering Method
  - The first step is to form a discretetization of the object space $\Re^S$ by forming a grid on $\Re^S$. The intersection of the grid lines, which occurs at what are called the vertex or node points, will provide our desired discretetization. The finite subset of $\Re^S$ consisting of the vertices is denoted as V. The set of points in V constitutes the candidates for cluster centers. Thus the degree of approximation of final centers is very sensitive to the fineness of the grid. The finer the grid the less approximate is the result but, more calculations are needed. The grid need not be uniform throughout the space $\Re^S$; different parts of the space may have a different resolutions of the grid.
  - The second step is the introduction of the data and construction of the mountain function. The mountain function denoted by

M is defined on V and is constructed as follows. For each piece of data, $\mathbf{x}_i$, an amount is added to the M value at each point $\mathbf{v} \in V$. The amount added depends on the distance of $\mathbf{v}$ from $\mathbf{x}_i$; the closer the two the more is added. In this way after all the data points are considered we get a function on $\Re^s$, actually V, which looks like a mountain range reflecting the distribution of the data. The next step is the selection of the cluster centers. This is accomplished by the destruction of the mountains. We find the point in V, $\mathbf{c}_1$, which has the greatest value for M, the peak of the mountain range, this becomes our first cluster center. For all points $\mathbf{v}$ in V we subtract from their M value a quantity dependent upon its distance from $\mathbf{c}_1$ and the value $M(\mathbf{c}_1)$. The effect of this subtraction is to reduce the mountains. We next look for the new peak. This becomes our next cluster center, $\mathbf{c}_2$. We now use $\mathbf{c}_2$ and its value to further reduce our mountain function. This is continued in this manner until the mountain function is virtually destroyed. For high dimensional data, to get quality prototypes, computational overhead of Mountain Clustering Method becomes very high. To overcome this, Chiu[19] suggested the Subtractive Mountain Clustering Method, where each data points is used as a potential cluster center instead of the grid points. Next we discuss our proposed Structural Mountain Clustering Method.

2.5 Structural Mountain Clustering Method(SMCM)

The Structural Mountain Clustering Method is a modified form of Subtractive Mountain Clustering Method [18]. The data set used is:

$$X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\};\ \mathbf{x}_i \in \Re^P;\ i = 1, 2, \ldots, N.$$

In SMCM each data point is considered a potential cluster center, For each piece of data, an amount is added to the M(mountain potential) value at every data point including itself. The amount added depends on the *structural similarity* of $\mathbf{x}_j$ from $\mathbf{x}_i; i \neq j$; the structural similarity is obtained after aligning the data points using Best Molecular Fit Routine(BMF), the closer the similarity between the two the more is added. In this way at every data point we compute the mountain potential M using all other data points. The next step is the selection of the cluster centers. Like MCM we find the hexamer, $\mathbf{c}_1$, in the data set, which has the highest value for M. This becomes our first cluster center. We form the clusters using the cluster center and assigning

members to clusters which are having RMS distance less that 1 Å. Note that, in MCM we do not do that, but we discount the potential and find next cluster center. We then eliminate all members in the first cluster from our data set. We now recompute the potential and look for the new peak. This becomes our next cluster center, $\mathbf{c_2}$. We now use $\mathbf{c_2}$ and its members to further reduce our data set. This is continued in this manner until all points in the data set are assigned to clusters.

The algorithm is as follows:
Given the data set

$$X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\};\ \mathbf{x}_i \in \Re^P;\ i = 1, 2, \ldots, N.$$

We use the best Molecular Fit Algorithm between every pair

$$(\mathbf{x}_i, \mathbf{x}_j);\ i \neq j; i = 1, 2, \ldots, N;$$

to compute

$$RMS_{i,j};\ \forall i, j$$

Repeat while all hexamers are not assigned to clusters

1. Calculate the *potential* at each hexamer $\mathbf{x}_i$ using the formula,

$$M(\mathbf{x}_i) = \sum_{\mathbf{x}_j, j \neq i} \exp(-\alpha \cdot RMS_{i,j})$$

   where $RMS_{i,j}$ is the Root Mean Square distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ after alignment, α is a positive real value; $\alpha > 0$

2. The hexamer which has the highest *potential* is chosen as the final *building block* and a cluster is formed by adding the *building block* and hexamers which has RMS 1 Å from the *building block*. The hexamers added to cluster are removed from the hexainers list.

Choice of α may have effect on the clusters extracted. We experimented with different choices of α.

- Variant of Self Organizing Feature Map

The SOM method is a kind of vector quantization algorithm which has been widely used as a statistical tool for investigating data structures due to its excellent visualization properties for high dimensional complex data sets [20]. The SOM gives a mapping from the high-dimensional input data space $\Re^n$ into a low-dimensional (usually 1-

dimensional or 2-dimensional) grid of units preserving neighborhood relations in the input space so that data points lying nearby each other in the input space are mapped onto nearby-map units. In other words, SOM preserves the topology of input on the map. SOM also has the property of density matching. Every unit has a reference vector $\mathbf{m}_i \in \Re^n$. An input data vector $\mathbf{x} \in \Re^n$ is compared with all $\mathbf{m}_i$ typically using Euclidean distance; the minimum of the Euclidean distance $\|\mathbf{x} - \mathbf{m}_i\|$ defines the best-matching unit for the input data vector $\mathbf{x}$, which is signified by the subscript $\mathbf{b}$:

$$\|\mathbf{x} - \mathbf{m}_b\| = \min_i \|\mathbf{x} - \mathbf{m}_i\| \tag{2}$$

In this way, $\mathbf{x}$ is mapped onto the unit $\mathbf{b}$. The reference vectors $\mathbf{m}_b$ and its neighbors $\mathbf{m}_i$ are adjusted by a learning process as follows:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \eta(t) h_{bi}(t)(\mathbf{x}(t) - \mathbf{m}_i(t)), \tag{3}$$

where t is the discrete time index and $h_{bi}(t)$ is a neighborhood function. Of the various choices, the Gaussian function is most popular one.

$$h_{bi}(t+1) = \exp\left(-\frac{\|\mathbf{r}_b - \mathbf{r}_i\|^2}{2\sigma^2(t)}\right), \tag{4}$$

is widely used, where $\mathbf{r}_b$ and $\mathbf{r}_i$ are the positions of best matching unit $\mathbf{b}$ and its neighboring units on the grid, $\eta(t)$ is the learning rate, and $\sigma(t)$ defines the width of the kernel. $\alpha(t)$ and $\sigma(t)$ decrease monotonically with time as follows:

$$\sigma(t+1) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right), \tag{5}$$

where σ_0 is the value of σ at the initiation of the SOM algorithm, and τ_1 is the time constant.

$$\eta(t+1) = \eta_0 \exp\left(-\frac{t}{\tau_2}\right), \tag{6}$$

where η_0 is the value of η at the initiation of the SOM algorithm, and τ_2 is another time constant.

The SOM Algorithm randomly selects a training data point and applies 3. This is called a step, When the algorithm is repeated enough number of steps, the map is expected to preserve topology of input data. Although SOM has been used in numerous application, it is not

useful for applications that need quantization of structures or shapes. In other words, if we want for example, helix structures of different orientation to map to the same node on the feature map, usual SOM will not work. So we propose a modification.

2.6 Structural Self Organizing Map (SSOM)

We shall illustrate and describe the algorithm using hexamers, although it can be used in other applications.
Algorithm SSOM: Input vector is a hexamer which is of dimension $p = 6 * 3 = 18$ consisting of 3D coordinates of C_α atom of six residues.

Consider a net of $P \times Q = l$ neurons

1. *Initialization.* The algorithm can be initialized by randomly generated l vectors from the smallest hyperbox in $\Re^p$ containing the training data. For a quicker convergence and better results the net can also be initialized by randomly selecting the weight vectors $\{\mathbf{m}_j(0)\}_{j=1}^{l}$ from the available set of input vectors $\{\mathbf{x}_i\}_{i=1}^{N}$ for $j = 1, 2, ..., l$, where l is the number of neurons in the lattice and N is the number of inputs.

2. *Sampling.* Draw a random sample hexamer $\mathbf{x}$ from the input data; the vector $\mathbf{x}$ represents the activation pattern that is applied to the net of neurons.

3. *Similarity Matching.* Find the (best-matching) (winning) neuron i at time step t by using minimum RMS (equation 1) distance criterion.

$$i = \arg\min_j \{RMS_j\},$$

where RMS_j is the distance between the aligned $\mathbf{x}(t)$ and $\mathbf{m}_j, j = 1, 2, \ldots, l$

4. *Updating.* Adjust the synaptic weights vectors of all neurons within a spatial neighborhood of the winner by using the update formula

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + \eta(t) h_{j,i}(t)(\acute{\mathbf{x}}(t) - \mathbf{m}_j(t))$$

where $\acute{\mathbf{x}}(t)$ is aligned version of $\mathbf{x}(t)$ corresponding to $\mathbf{m}_j(t)$ using BMF routine, where t is the discrete time index and $h_{j,i}(t)$ is the neighborhood function. We use the Gaussian function

$$h_{j,i}(t+1) = \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_j\|^2}{2\sigma^2(t)}\right), \tag{7}$$

where $\mathbf{r}_i$ and $\mathbf{r}_j$ are the positions of the winner neuron i and the neuron j in its neighborhood. Like usual SOM $\eta(t)$ is the learning rate, and $\sigma(t)$ defines the width of the kernel. $\alpha(t)$ and $\sigma(t)$ decrease monotonically with time as following

$$\sigma(t+1) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right), \tag{8}$$

where σ_0 is the value of σ at the initiation of the SOM algorithm, and τ_1 is the time constant.

$$\eta(t+1) = \eta_0 \exp\left(-\frac{t}{\tau_2}\right), \tag{9}$$

where η_0 is the value of η at the initiation of the SOM algorithm, and τ_2 is another time constant.

5. *Continuation.* Following a suggestion by Kohonen [20], The updating of weights is continued for 500 times the number of neurons($500 \times P \times Q$ *steps*).

6. For finer refinements, the network is trained for some steps with only winner update strategy.

3 Application of SSOM to Protein Data

3.1 Characteristics of SSOM

SSOM can find interesting clusters that can be useful directly to pattern recognition type application. But here we have an additional constraint. We want to limit the quantization error for each hexamer in the training data to 1 Å. So we used a second stage to recluster a cluster obtained by SSOM, if the dispersion in the cluster is more than what is desired.

It consists of two stages, in the first stage SSOM algorithm is applied to get major clusters, and in second stage subclustering of the major clusters is done by using SMCM on each major cluster which has members with RMS deviation more than 1Å. We have also used the two-stage clustering of [1] for subclustering of the major SSOM clusters.

4 Reconstructing proteins by standard building blocks

4.1 Method of Reconstruction

Following [1] we use the following procedure: First, we replace each original hexamer of the protein by its closest (in terms of RMS distance) standard building block. Then, since the building blocks overlap, we align every two consecutive building blocks by using the BMF algorithm. Onto the suffix (the last five residues) of the first building block we fit the prefix (the first five residues) of the next building block. Thus, the 3D position of the last residue of the latter hexamer is determined and is added to the growing chain. This process is repeated until the whole protein is reconstructed.

5 Results

5.1 Performance of Building Blocks

In order to estimate how well these building blocks represent hexamers found in proteins, we tested each of the 12,973 hexamers in our refined Brookhaven data base. We enumerate our findings below:

1. *Two Stage Clustering* [1] The 406 hexamers were first clustered into 55 distinct clusters. These clusters were then further subdivided as described earlier to give a total of 103 subclusters. The central hexamer of each subcluster was selected to be a standard building block. 76% of hexamers had a distance of less than 1 Å from at least one of the standard building blocks, and 92% had a distance of less than 1.25Å from one of them. The average distance between a hexamer and its closest building block (including the 24% of the hexamers whose closest building block was at a distance greater than 1 Å) was 0.74Å. Table 2 depicts 30 most popular building blocks found and the corresponding number of hexamers which are at a distance of less than 1 Å.

2. *Structural Mountain Clustering Method*
   Using $\alpha = 3.3$, we got 105 clusters. Using these 105 building blocks we find that 77.1% of hexamers had a distance of less than 1 Å from at least one of the standard building blocks, and 92.6% had a distance of less than 1.25Å from one of them. The average distance between a hexamer and its closest building block (including the 23.9% of the hexamers whose closest building block was at a distance greater than 1 Å) was 0.74 Å. The top 30 building block found by [1] can model 7716 hexamers(see Table 2) while the top 30 building blocks obtained by SMCM can model 9515 hexamers(see Table 3). So SMCM performs better than the method in [1].

3. *Structural Self Organizing Map*
   Applying SSOM, the 406 hexamers were first clustered into 49 major clusters. The number of clusters depends on the number of neurons in the lattice. The fact that our SSOM algorithm did not produce a small number of large clusters showed that indeed the structure of hexamers does not vary gradually and they are readily separable into distinct clusters. These clusters were then further subdivided using SMCM to give 130 subclusters. The central hexamer of each subcluster was selected to be a standard building block. In this case 81.3% of hexamers have a distance of less than 1 Å from at least one of the standard building blocks, and 94.3% had a distance of less than 1.25 Å from one of

Table 2: 30 Most Popular Building Blocks obtained from Two Stage Clustering [1]

No.	Protein	Residue	Sequence	No. of occurrences
1	1BP2	104	ICFSKV	2908
2	1BP2	80	NEITCS	901
3	5PTI	31	QTFVYG	403
4	1BP2	57	KLDSCK	319
5	1BP2	108	KVPYNK	286
6	1BP2	74	SYSCSN	283
7	4HHBb	54	VMGNPK	226
8	4HHBb	2	HLTPEE	194
9	1PCY	40	VFDEDS	151
10	4HHBb	1	VHLTPE	148
11	4HHBb	82	KGTFAT	145
12	4HHBb	16	GKVNVD	135
13	5PTI	30	CQTFVY	135
14	1PCY	85	SPHQGA	132
15	1BP2	18	PLLDFN	132
16	1PCY	86	PHQGAG	119
17	1PCY	10	GSLAFV	116
18	5PTI	27	AGLCQT	100
19	1BP2	84	CSSENN	99
20	1BP2	61	CKVLVD	95
21	5PTI	5	CLEPPY	91
22	4HHBb	32	LVVYPW	88
23	1PCY	72	VALSNK	88
24	1PCY	2	DVLLGA	88
25	1PCY	66	KGETFE	85
26	1BP2	30	GLCGSG	85
27	1PCY	7	ADDGSL	83
28	1BP2	56	KKLDSC	73
29	4HHBb	95	KLHVDP	72
30	1PCY	53	SKISMS	68
				7716

Table 3: 30 Most Popular Building Blocks obtained from Mountain Clustering

No.	Protein	Residue	Sequence	No. of occurrences
1	1BP2	50	NCYKQA	2849
2	1BP2	80	NEITCS	902
3	1PCY	71	EVALSN	661
4	5PTI	20	RYFYNA	540
5	5PTI	28	GLCQTF	340
6	4HHBb	49	STPDAV	288
7	1PCY	17	SEFSIS	247
8	1PCY	50	VDASKI	234
9	5PTI	39	RAKRNN	227
10	4HHBb	53	AVMGNP	211
11	5PTI	33	FVYGGC	199
12	4HHBb	91	LHCDKL	197
13	4HHBb	118	FGKEFT	194
14	1PCY	23	PGEKIV	192
15	1BP2	82	ITCSSE	179
16	1PCY	28	VFKNNA	177
17	1PCY	86	PHQGAG	172
18	1PCY	49	GVDASK	157
19	5PTI	14	CKARII	154
20	1PCY	86	SPHQGA	149
21	5PTI	7	EPPYTG	139
22	1BP2	85	SSENNA	133
23	4HHBb	95	KLHVDP	129
24	4HHBb	36	PWTQRF	128
25	1PCY	2	DVLLGA	123
26	1PCY	83	YCSPHQ	123
27	1PCY	91	GMVGKV	123
28	1PCY	11	SLAFVP	116
29	1PCY	10	GSLAFV	116
30	1PCY	65	AKGETF	116
				9515

them. The average distance between a hexamer and its closest building block (including the 18.7% of the hexamers whose closest building block was at a distance greater than 1 Å) was 0.69 Å. So the performance of the building blocks are much better than those by [1] and SMCM. One may argue that, it uses more building block and hence better result. Analyzing Table 4, which shows the performance of top 30 building blocks, we find that they can model 41929 hexamers, which is much higher than the performance of top 30 building blocks obtained in [1] or SMCM. This indicates that there are some very good prototypes and there are some poor prototypes/building blocks generated by the SSOM based system. This may be attributed to the density matching property of SOM. In a high dense area, it places more prototypes giving better quantization, and after applying 1 Å threshold, few points may be left out which are treated as prototypes.

We also applied the two-stage method of [1] for subclustering of SSOM clusters. The performance of the top 30 building blocks found by this method is shown in Table 5. Performance of this system is also much better than [1].

5.2 Performance of Reconstruction

Evaluating the performance of reconstruction procedure is not a trivial task, following the methods in [1] only the first 60 residues of each protein were used in order to have a standard (not too long) protein length on which to test our approach. From the refined Brookhaven data base, all of the proteins (71) of length greater than 60 are used and only their first 60 residues are used. One of the simplest ways to measure the similarity between the original and reconstructed structure is to calculate the RMS distance between them. These distances were compared to the average "random" distance between structures, which we took to be the average distance measured between any pair of truncated proteins in our library. Thus, we had 2485 pairs (from the 71 proteins of length 60) with average distance of 12.85 A and SD of 2.12 A.

1. *Two Stage Clustering [1]* The average distance between the original proteins and the reconstructed ones is 7.3 Å, which is 2.6 SD lower than the random average. Of the proteins 28% had been reconstructed with an RMS of less than 5 Å,but 25% had RMS distances greater than 9 Å.

Table 4: 30 Most Popular Building Blocks obtained from SSOM after sub-clustering by SMCM

No.	Protein	Residue	Sequence	No. of occurrences
1	1BP2	104	ICFSKV	2908
2	4HHBb	125	PVQAAY	2881
3	4HHBb	134	VAGVAN	2859
4	4HHBb	126	VQAAYQ	2853
5	4HHBb	133	VVAGVA	2850
6	4HHBb	89	SELHCD	2849
7	4HHBb	65	KKVLGA	2847
8	4HHBb	137	VANALA	2823
9	4HHBb	128	AAYQKV	2819
10	4HHBb	19	NVDEVG	2805
11	4HHBb	114	LAHHFG	2674
12	5PTI	2	PDFCLE	1419
13	1PCY	94	GKVTVN	989
14	1PCY	93	VGKVTV	885
15	1PCY	26	KIVFKN	882
16	4HHBb	38	TQRFFE	748
17	4HHBb	99	DPENFR	730
18	1PCY	36	PHNIVF	540
19	1PCY	18	EFSISP	536
20	4HHBb	139	NALAHK	484
21	1BP2	72	NYSYSC	462
22	5PTI	32	TFVYGG	444
23	5PTI	17	RHRYF	442
24	1PCY	81	SFYCSP	420
25	5PTI	51	CMRTCG	405
26	5PTI	15	KARHR	381
27	5PTI	28	GLCQTF	340
28	1PCY	56	SMSEED	226
29	5PTI	38	CRAKRN	216
30	4HHBb	1	VHLTPE	212
				41929

Table 5: 30 Most Popular Building Blocks obtained from SSOM after Sub-clustering by method [1]

No.	Protein	Residue	Sequence	No. of occurrences
1	4HHBb	89	SELHCD	2849
2	4HHBb	65	KKVLGA	2847
3	5PTI	49	EDCMRT	2843
4	4HHBb	86	ATLSEL	2840
5	5PTI	50	DCMRTC	2831
6	4HHBb	128	AAYQKV	2819
7	4HHBb	19	NVDEVG	2805
8	4HHBb	114	LAHHFG	2674
9	4HHBb	71	FSDGLA	2666
10	5PTI	2	PDFCLE	1419
11	1PCY	94	GKVTVN	989
12	4HHBb	38	TQRFFE	748
13	1PCY	71	EVALSN	661
14	1PCY	18	EFSISP	536
15	4HHBb	139	NALAHK	484
16	5PTI	32	TFVYGG	444
17	1PCY	80	YSFYCS	437
18	5PTI	31	QTFVYG	408
19	5PTI	51	CMRTCG	405
20	5PTI	15	KARHR	381
21	5PTI	46	KSAEDC	326
22	1PCY	17	SEFSIS	247
23	4HHBb	14	LWGKVN	247
24	1BP2	79	NNEITC	246
25	1PCY	5	LGADDG	235
26	4HHBb	54	VMGNPK	228
27	5PTI	39	RAKRNN	227
28	1PCY	56	SMSEED	226
29	5PTI	38	CRAKRN	216
30	4HHBb	1	VHLTPE	212
				34496

2. *Structural Mountain Clustering Method*
   The average distance between the original proteins and the reconstructed ones is 7.2 Å, which is 2.64 SD lower than the random average. Of the proteins 32% had been reconstructed with an RMS of less than 5 Å,but 24% had RMS distances greater than 9 Å. So SMCM marginally outperforms [1].

3. *Structural Self Organizing Map*
   The average distance between the original proteins and the reconstructed ones is 6.9 Å, which is 2.9 SD lower than the random average. Of the proteins 35% had been reconstructed with an RMS of less than 5 Å,but 20% had RMS distances greater than 9 Å. Thus, SSOM outperforms both methods in [1] and SMCM.

6 Conclusion and Discussion

6.1 Conclusion

We proposed two new schemes named the *Structural Self Organizing Map method* and *Structural Mountain Clustering method* for finding building blocks or prototypes in a database of structures. The two proposed methods are tested on bench mark data sets and are found to produce good prototypes than the Two Stage Clustering Method [1]. The Structural Mountain Clustering Method is simple and finds prototypes in one stage, unlike Two stages of clustering method. The Structural Mountain Clustering Method yields prototypes that match to more members in database of hexamers within threshold of 1 Å. The prototypes extracted also produce good reconstructed protein 3D structure. Similarly, the building blocks extracted by the two SSOM based schemes are also quite good both for representation of database and reconstruction of protein structure.

One of the aim of this thesis was to show the existence of a manageable-sized set of standard building blocks that has sufficient expressive power to replace most of the oligomers in known structures. We can thus answer first question in the affirmative; the fact that near about 50 disjoint clusters were formed indicates that the known hexamers can be divided into distinct structural motifs. These structural motifs include the well-known types of secondary-structure elements with finer resolution, and many standard units that connect them. The classification into a set of a few dozen building blocks seems to be more meaningful than the crude classification to very

few secondary-structure elements. It is stressed that building blocks are not secondary structure elements. Even if the secondary structure assignment of a protein is known, it is not clear how to assign three-dimensional structure to the protein. However, because our building blocks reflect three-dimensional information (for example, the direction of a turn after a helix) they easily lend themselves to the reconstruction process. Thus, we can give a positive answer to the second question: The simple reconstruction algorithm that we applied yields good results in simulating the original proteins.

6.2 Discussion

6.2.1 Why hexamers?

In this thesis we concentrated on an analysis of hexamers as building blocks. It is clear that there is no magic in the number six. We considered hexamers as a starting point for our research for the following pragmatic reasons:

- Kabsch and Sander [11] observed that the same pentamer sequence can be found in totally different conformation in different proteins. Thus, pentamers by themselves seem to be too short to carry structural stability. The length of secondary structure elements is usually in the range of 4-16, e.g., the length of a turn is usually 4-6 amino acids; helices and sheet have larger ranges but they are still usually less than 16 residues.

- When using longer oligomers, their sequence dependency seems to fade out. Hence, longer building blocks may be less sequence specific, and we should use as short fragments as possible.

6.2.2 Prediction of 3D structures of Proteins with Unknown 3D structures

Each building block is associated with a sequence distribution matrix. This matrix can be obtained simply by counting, for all of the hexamers in the cluster represented by this building block. The matrix gives the distribution of amino acids at each one of the six positions along the hexamers. These matrices can be normalized against the distribution of the different amino acids in the data base. These matrices reflect the sequences of the hexamers that are represented by each building block. They can be used to assign a hexamer sequence to its corresponding building block. The sequence of each hexamer is matched against these matrices and assigned to the building block whose matrix it fits the best. Then the construction procedure described in

the section Reconstruction of Protein Structure can be applied for the protein sequence with unknown 3D structure to predict its 3D structure.

References

[1] Ron Unger, David Harel, Scot Wherland, and Joel L. Sussman. A 3d building blocks approach to analyzing and predicting structure of proteins. *PROTEINS: Structure, Function, and Genetics*, 5:355–373, 1989.

[2] World Wide Web. `http://www.rcsb.org/pdb`.

[3] C. B. Anfinsen and Edgar Haber. Studies on the reduction and reformation of protein disulfide bonds. *J. Biol. Chem*, 236:1361–1363, 1960.

[4] C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.

[5] Chuen-Der Huang, Chin-Teng Lin, and Nikhil Ranjan Pal. Hierarchical learning architecture with automatic feature selection for multiclass protein fold classification. *IEEE Transactions on Nanobioscience*, 2(4), December 2003.

[6] C Bystroff and D Baker. Prediction of local structure in proteins using a library of sequence-structure motifs. *J. Mol. Biol.*, 281:565–577, 1988.

[7] Carl-Ivar Branden and John Tooze. *Introduction to Protein Structure: 2nd edition*. Garland Publishing, New York, 2nd edition, 1999.

[8] Y. Liu and D. L. Beveridge. Exploratory studies of ab initio protein structure prediction:multiple copy simulated annealing, amber energy functions, and a generalizedborn/solvent accessibility solvation model. *Proteins*, 46, 2002.

[9] R. Unger and J. Moult. A genetic algorithm for 3d protein folding simulations. In *5th Proc. Intl. Conf. on Genetic Algorithms*, pages 581–588, 1993.

[10] P. Pokarowski, A. Kolinski, and J. Skolnick. A minimal physically realistic protein-likelattice model: Designing an energy landscape that ensures all-or-none folding to a unique native state. *Biophys J*, 84:1518–26, 2003.

[11] W. Kabsch and C. Sander. On the use of sequence homologies to predict protein structure: Identical pentapeptides can have completely different conformations. *Proc. Natl. Acad. Sci. U.S.A.*, 81:1075–1078, 1984.

[12] LA. Wilson, D.H. Haft, E.D. Getzoff, J.A. Tainer, R.A. Lerner, and S. Brenner. Identical short peptides in unrelated proteins can have different conformations: A testing ground for theories of immune recognition. *Proc. Natl. Acad. Sci. U.S.A.*, 82:5255–5259, 1985.

[13] P. Argos. Analysis of sequence-similar pentapeptides in unrelated protein tertiary structure: Strategies for protein folding and a guide for site-directed mutagenesis. *J. Mol. Biol.*, 197:331–348, 1987.

[14] M. Rooman and S.J. Wodak. Identification of predictive sequence motifs limited by protein structure data base size. *Nature (London)*, 335:45–49, 1988.

[15] S.C. Nyburg. Some uses of a best molecular fit routine. *Acta Crystallogr.*, B30:251–253, 1974.

[16] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr.*, B32:922–923, 1976.

[17] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr.*, A34:828–829, 1978.

[18] Approximate Clustering Via the Mountain Method. Ronald r. yager and dimitar p filev. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 25(8):1279–1284, AUGUST 1994.

[19] S.L. Chiu. Fuzzy model identification based on cluster estimation. *J. Intelligent Fuzzy Systems*, 2(267-278), 1994.

[20] T. Kohonen. Self-organizing maps. *Springer, Berlin and Heidelberg, Germany,*, 1995.