STRUCTURAL ALIGNMENT USING NETWORK PROPERTIES

by

GALİP GÜRKAN YARDIMCI

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
February 2007

STRUCTURAL ALIGNMENT USING NETWORK PROPERTIES

APPROVED BY:

Assoc. Prof. Dr. Uğur Sezerman (Thesis Advisor) .....

Assoc. Prof. Dr. Hikmet Budak .....

Assist. Prof. Dr. Devrim Gözüaçık .....

Assist. Prof. Dr. Yücel Saygın .....

Prof. Dr. Zehra Sayers .....

DATE OF APPROVAL : .....

STRUCTURAL ALIGNMENT USING NETWORK PROPERTIES

Galip Gürkan YARDIMCI

BIO, M.S. Thesis, 2007

Thesis Supervisor: Assoc. Prof. Dr. Uğur SEZERMAN

Keywords: Proteins, alignment, graph theory, contact map, structural domains

Abstract

Understanding the structural means of protein function via structural comparisons have wide range of applications such as protein fold classification, protein structure modelling and design. In this thesis, a novel structural alignment algorithm based on a amino acid network model is presented.

The method we present models proteins as an amino acid network, derived from contact map representation of proteins. By using this model, we obtain fast tertiary structure comparisons, and combine them with primary and secondary structure comparisons to develop an overall similarity function. The similarity function drives a dynamic programming based alignment algorithm to obtain fast and accurate structural alignments.

The structural alignments obtained are used to discover functional structural subunits called domains and to discover overall structural similarity of two proteins. We compared our domain prediction results with existing domain recognition methods and saw that our method correlates well with existing methods. Our global structural alignment results are compared with CE alignments.

AĞ ÖZELLİKLERİ KULLANARAK YAPISAL HİZALAMA

Galip Gürkan YARDIMCI

BIO, Yüksek Lisans Tezi, 2007

Tez Danışmanı: Doçent. Dr. Uğur SEZERMAN

Anahtar Kelimeler: Proteinler, hizalama, grafik teorisi, değme haritasi, yapısal kümecikler

Özet

Proteinin işlevinin yapısal karşılaştırma kullanarak anlaşılmasının proteinin katlanma sınıflandırılması, protein yapı modellemesi ve tasarımı gibi farklı alanlarda uygulamaları vardır. Bu savda, amino asit ağ modeline dayalı yeni bir yapısal hizalama algoritması sunulmaktadır.

Sunulan yöntem değme haritası gösterimden yararlanarak proteinleri amino asit ağları olarak modellemektedir. Bu modeli kullanarak hızlı üçüncü yapı karşılaştırmaları elde edilir ve birinci ve ikinci yapı karşılaştırmaları ile birleştirilerek genel bir benzerlik fonksiyonu elde edilir. Benzerlik fonksiyonu dinamik programlamaya dayalı bir hizalama algoritmasını yönlendirerek hızlı ve doğru hizalamalar elde etmeyi sağlar.

Elde edilen yapısal hizalamalar işlevsel yapısal altbirimler olan yapısal kümecikleri ve genel yapısal benzerliği keşfetmekte kullanılır. Yapısal kümecik tahmini sonuçlarımız varolan diğer yapısal kümecik tanımlama yöntemleri ile karşılaştırıldı ve aralarında uygunluk olduğu görüldü. Genel yapısal hizalama sonuçlarımız CE hizalamaları ile karşılaştırıldı.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVATIONS

PDB: Protein Data Bank

PAM: Point Accepted Mutations

BLOSUM: BLOcks SUbstitution matrix

RMSD: Root Mean Squared Deviation

CE: Combinatorial Extension

DALI: Distance Matrix ALIgnment

CATH: A Hierarchic Classification of Protein Domain Structures

SCOP: Structural Classification of Proteins

C_α: Alpha Carbon

C_β: Beta Carbon

c: connectivity

ς: cliquishness

1 INTRODUCTION

The development of high throughput experimental methods in molecular biology has caused the number of discovered proteins increase on a yearly basis. With the aid of bioinformatics, these proteins are stored in online databases, made accessible to all via Internet. This ever increasing huge amount of data, stored in databases like SWISS-PROT, GENBANK and PDB, needs to be analyzed and classified accordingly to facilitate further growth, ease of use and information retrieval.

For analysis and classification purposes, a basic requirement is a distance measure or a similarity assessment. Protein alignments, sequential or structural, are widely used and accepted methods to discover similar regions between proteins and to assess the similarity by a score. Especially structural alignment methods, which are capable of capturing structural thus functional homologies, are useful tools for protein fold classification, protein structure modeling and structure based annotation. With rapidly growing databases, the need for fast and accurate structural alignment algorithms is apparent.

In this thesis, we try to find structurally and sequentially similar regions between homologous proteins. Homologous proteins contain structural elements called domains, which have unique structures and sequences. For some domains, amino acid sequences may not be unique, but the structure always exhibit strong similarity, which also gives the domain its specific function. Thus, we further expect that we'll be able to discover the domains by aligning proteins if both proteins possess the same domain. Domains are self independently folding structural units, each capable of fulfilling a specific duty, and many protein chains contain one or more domain structures that make the functioning of the

protein possible. Discovery of domains can also aid synthetic multifunctional protein design by using domains as building blocks.

The alignment process is based on a dynamic programming approach to discover the best alignment between two proteins. The similarities are assessed by a special function that combines sequence similarity and structural similarity in such a way that two types of measures consolidate each other. We model each protein as an amino acid network represented by a contact map, and assess the structural similarity by comparing connectivity and cliquishness parameters of nodes (amino acids) in the graph. These two parameters represent the local contact relationships between amino acids. In addition to this structure parameter, we also use the secondary structure similarity to assess the matches between different secondary structure elements. We expect that homologous proteins preserve local topologies and contacts, and the similarities between them can be captured by comparing connectivity, cliquishness and secondary structure parameters.

1.1 Organization of Thesis

Chapter 2 presents the biological background of our study and the literature survey of the existing methods addressing the issues addressed in this thesis. The fundamental idea of an alignment is explained as well. In Chapter 3, we explain our method in detail and explain the concepts used in developing our method. In Chapter 4, the development of the thesis is shown, in terms of experiments. Each experiment is presented with the set of results it yields and discussion of the results. In Chapter 5, the conclusion is made with a short summary, a general discussion of the method and future directions.

2 BIOLOGICAL BACKGROUND AND LITERATURE SURVEY

In this section the biological information related to the concepts and ideas presented in thesis will be explained. The data unit of our algorithm is a protein, thus we start with the proteins.

2.1 Proteins

Proteins are organic compounds that are vital to all organisms. Proteins are composed of linear chains of amino acid molecules. The sequence of the amino acids in a certain protein is determined by the DNA sequence of the gene that encodes that protein. All proteins have unique sequences, and different sequences provide the wide range of diverse functions proteins fulfill.

Proteins are synthesized in the cell by ribosomes. Ribosomes attach amino acids one after another in the order dictated by the gene that codes for that protein, forming the amino acid chain that will be the protein. Proteins fold into their three dimensional structure during synthesis, and the structure determines the function of the protein. This structure is determined by the interaction of the amino acids. To understand how the proteins function and rules that govern the protein folding, types and properties of amino acids will be covered first.

2.1.1 Amino acids

Amino acids are the building blocks of proteins. Amino acid is a molecule that contains an amine (NH_2) group, a carboxyl (COOH) group, a hydrogen atom and a side

chain attached to a carbon atom which is referred as alpha carbon (C_α) [1]. There are twenty different types of amino acids, the type of the amino acid is determined by the composition of the side chain.

The carboxyl and amine groups can react by releasing a water molecule to form a special bond called peptide bond. Long chains of amino acids are linked by peptide bonds formed between succeeding amino and carboxyl groups. Such chains are called peptides or poly-peptides. Side chain group does not take part in a peptide bond [2].

A side chain can be formed of different atom groups and defines the physical and chemical properties of an amino acid (Fig.2.1). For instance, if the amino acid is charged, the charge is on the side chain atoms. Amino acids can be classified into groups by different schemes based on side chain properties. One widely used classification is based on polarity of the side chain, it determines whether an amino acid is hydrophilic or hydrophobic. Hydrophilic amino acids can be divided into two groups, polar amino acids and charged amino acids.

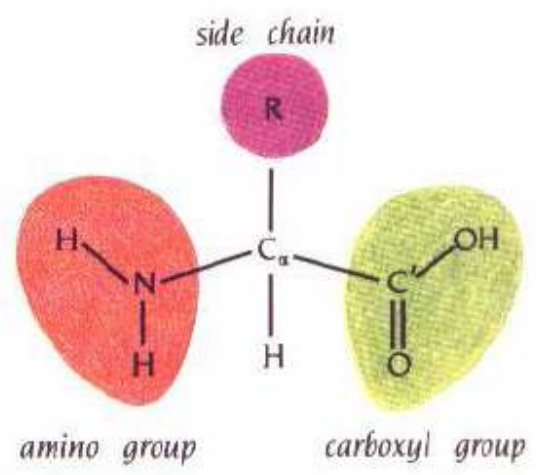

Figure 2. 7 Atomic structure of amino acid [1]

Hydrophobic amino acids do not favor to contact water molecules as their name implies. They mostly consist of carbon atoms, that's why they shy away from making contact with water. Amino acids of this type tend to be buried in the core of a protein to

avoid being close to water molecules surrounding the protein. This inclination is one of the major factors that determine the three dimensional structure of proteins. Hydrophobic amino acids are alanine, proline (weakly hydrophobic with small non-polar side chains), valine, leucine, isoleucine, phenylalanine and methionine (strongly hydrophobic with large side chains). Another property of proline and phenylalanine (also tryptophane and tyrosine which are polar) is that they are aromatic amino acids, meaning that their side chain form a ring structure. Other hydrophobic groups are called aliphatic amino acids because they are established by different combinations of CH_3 groups [1].

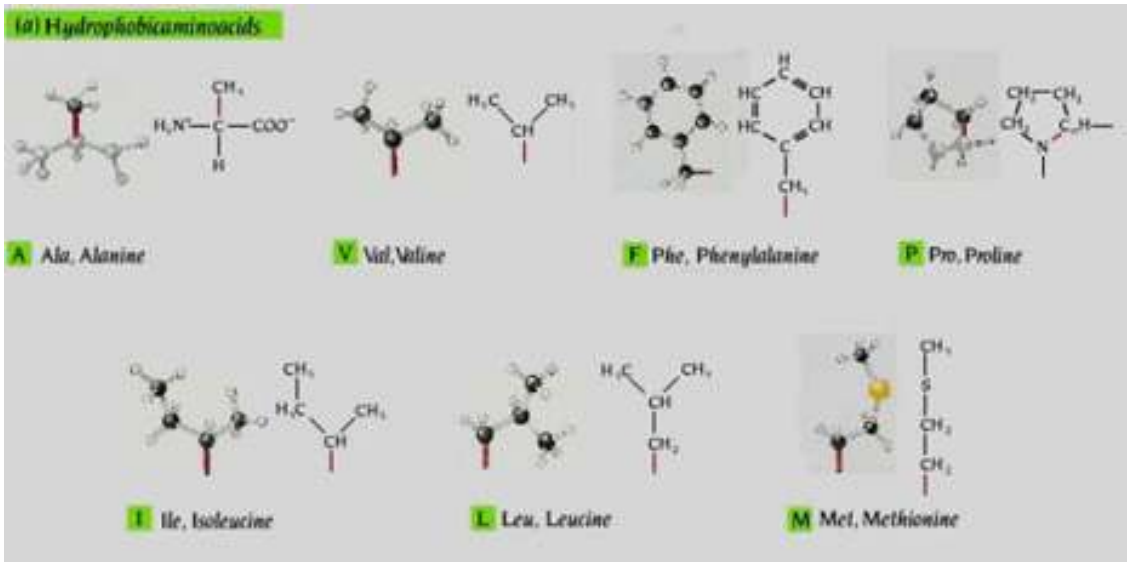

Figure 2. 8 Hydrophobic amino acid side chains [2]

Charged amino acids are usually found on the surfaces of proteins. They interact with water, opposite charged groups or other molecules proteins are designed to bind. The charge is on the atoms in the side chain. Aspartic acid and glutamic acid contain carboxyl groups in their side chains so they are negatively charged amino acids. Side chains of lysine and arginine contain amine groups, which gives them positively charge. Charged amino acids can make salt bridges, ionic bonds between positively charged and negatively charged, which contributes to protein stability [1].

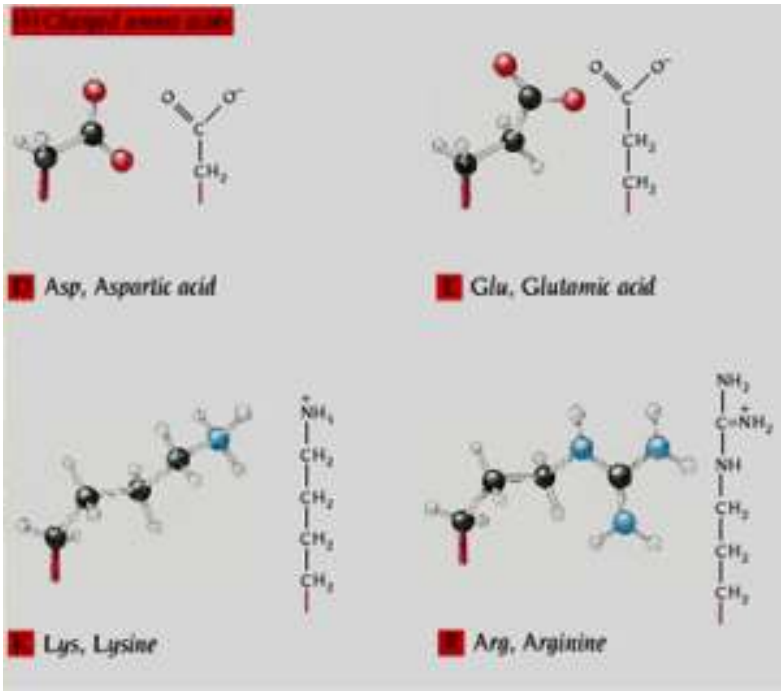

Figure 2. 9 Charged amino acid side chains [2]

Polar amino acids are neutral and their side chains are more soluble than non-polar amino acids (more hydrophilic) because they contain functional groups that form hydrogen bonds with water molecules. Therefore, they can be found on the exterior or interior regions of proteins. Polar amino acids are serine, threonine, tyrosine (both have side chains with hydroxyl group that makes hydrogen bonds), asparagine, glutamine, histidine (histidine may be neutral or positively charged based on the pH of the environment), tryptophan, glycine (glycine's side chain is a single hydrogen atom), cysteine (side chain contains a sulphur atom which can from stabilizing disulphide bonds) [1].

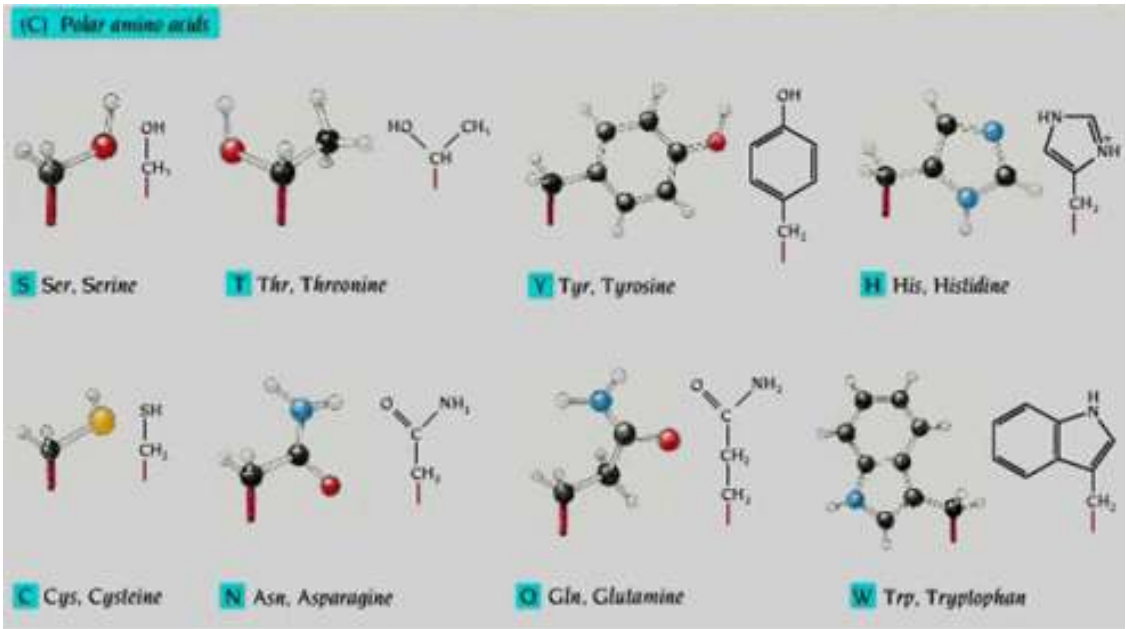

Figure 2. 10 Polar amino acid side chains [2]

Different types of amino acids are linked by peptide bonds to form the poly-peptide chain that will be the protein. After the poly-peptide chain is formed, it takes a certain shape which allows it to be a functional protein.

2.2 Protein Synthesis and Folding

Each protein is formed by a specific sequence of amino acids and sequence information is stored as genes in the DNA of each cell. Protein synthesis takes place in cytoplasm thus the sequence information must be transferred from the DNA in nucleus to cytoplasm. The sequence information is copied onto messenger RNA molecules (m-RNA), this process is called transcription. After the transcription, mRNA molecules are transferred to cytoplasm.

In the cytoplasm, ribosome organelles process the data on the mRNA molecules and attach the amino acids one after other, making a poly-peptide chain in the order specified by mRNA. This process is called translation. However translation is not the final step.

After translation, a protein assumes a certain stable conformation that minimizes its free energy, and this conformation is the conformation protein assumes for most of its activity. This process is largely dependent on the amino acid composition of the protein. The shape taken by the protein is also the shape that allows the protein to fulfill its duties. The shape dependent function is what makes proteins an irreplaceable element of metabolisms of all organisms.

The conformation of a protein is determined by certain forces. The electrostatic and covalent bonds between atoms are a major factor in the determination of conformation, such bonds are salt bridges, hydrogen bonds and disulfide bonds. Salt bridges are formed between positively and negatively charged atoms. Hydrogen bonds occur between a hydrogen atom and an electronegative atom, and occur a lot among atoms of amino acids that form alpha helices and beta sheets. Disulfide bonds occur between two sulfur atoms. Disulfide bonds are the most stabilizing type of electrostatic bonds and occur between cysteines which contain a sulfur atom in the side chain group [4].

There are also other factors that act to determine a protein's conformation. The Van der Waals interactions between atoms have a significant effect. The Van der Waals interaction occurs between very close atoms, creating influences on electron clouds and resulting in a weak attraction between the atoms. Even though the effect of each interaction is small, since there are so many interactions, the total effect on conformation is important [2].

Proteins also try to bury hydrophobic residues in the protein core since these residues don't favor contact with water, while polar and charged are on surface of the protein to make them accessible to water. This also promotes the formation of hydrogen bonds between water and charged/polar amino acids.

The sum of all these factors, weak factors (hydrogen bonds, hydrophobic effect and Van der Waals interactions) and others contribute to determine the stable folded

conformation of the protein [3]. These factors enable the protein to overcome entropic and enthalpic constraints in order to minimize its free energy of folded state.

Proteins try to minimize their free energy during folding process (a negative value of ΔG) because it enables them to become more stable. In equation (2.1), the free energy change is formulated in terms of enthalpy and entropy where ΔG is the free energy change between folded and unfolded state, ΔH is the enthalpy change and ΔS is the entropy change from folded to unfolded state. The enthalpy change, ΔH, corresponds to the binding energy (dispersion forces, electrostatic interactions, van der Waals potentials and hydrogen bonding) while hydrophobic interactions are described by the entropy term, ΔS.

$$\Delta G = \Delta H - T.\ \Delta S \tag{2.1}$$

Entropy expresses randomness or disorder of components of a system, and randomness is favored by nature. When the proteins fold and becomes ordered, this reduces the entropy, causing a negative value of ΔS, which does not favor folding. However the folding of the proteins is driven by the hydrophobic effect, burial of hydrophobic residues to core, and this causes the polar groups localization to protein surface. This gives more freedom to water molecules around the water, thus increases entropy.

Enthalpy (H) is the heat content of a chemical system. During folding, proteins maximize the hydrogen bonding between its own molecules, resulting in a release of energy, which in turn causes a negative ΔH. A negative ΔH favors bonding since it causes a negative ΔG, minimization of free energy.

2.3 Protein Structure

Protein structure is described in four levels. Each level describes protein structure at a different degree of complexity. The four structure levels are primary, secondary, tertiary and quaternary structures. [2]

In biochemistry, primary structure of a molecule is the exact atomic composition and the bonds connecting the atoms of the molecule. This definition has been generalized for proteins since all proteins are connected by peptide bonds and amino acid itself suggests the atomic composition. For a protein, the primary structure is the amino acid sequence of the protein.

Secondary structure describes the general three dimensional configuration of the local regions of a polymer. Hydrogen bonds among the amino acids are defining factors for secondary structure because they signify local contacts, therefore local form. Two basic secondary structure units are alpha helix and beta sheet. These local structure conformations are held together by hydrogen bonds.

The alpha helix is a right handed coil conformation, its shape resembling a spring. Each amino acid has a 100 degree turn, so there are average 3.6 amino acids in a complete turn (Figure 2.2). In alpha helices, the amine group of an amino acid makes a hydrogen bond with the carboxyl group of an amino acid four ahead in the protein sequence, known as (i,i+4) bonding. This way, hydrogen bonds stabilize the helix in a parallel direction to helical axis. In this conformation, the side chains of amino acids are located on the outer region of the helix. [2]

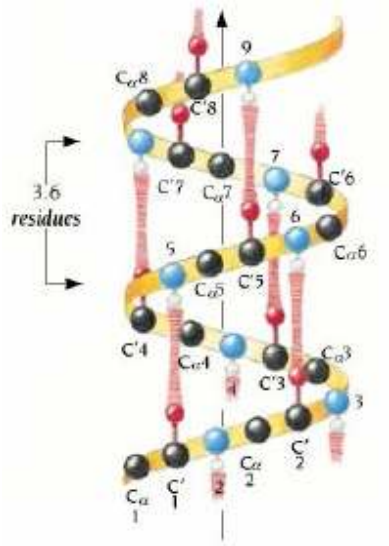

Figure 2. 11 Forming of an alpha helix [1]

The beta sheet consists of strands of amino acids connected to each other by hydrogen bonds (Fig.2.3). Amino acids in one chain make hydrogen bonds with other amino acids in neighboring strands, the hydrogen bonds are between the backbone carboxyl group of one amino acid and amine group of other. Beta sheets can be parallel or anti-parallel depending on the biochemical direction of neighboring strands. If the strands are in the same direction, the beta sheet is said to be parallel, else it's anti-parallel.

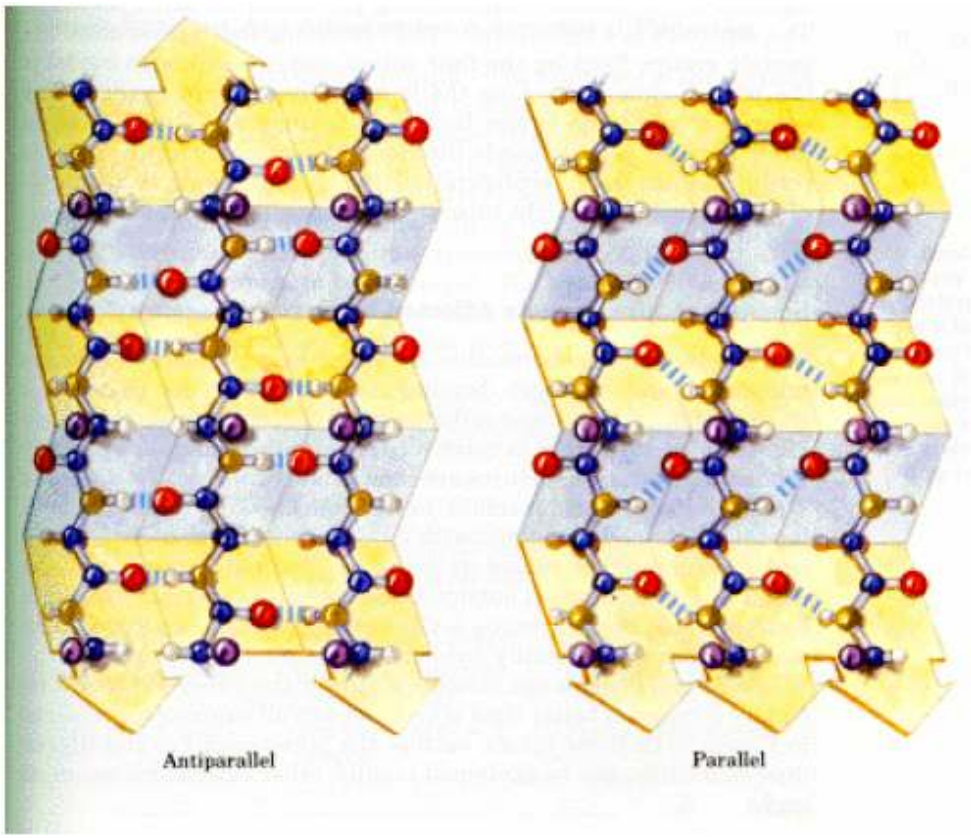

Figure 2. 12 Beta sheet structure [1]

The third secondary structure element is the loop. The loop is defined by the close approach of two amino acids when these two amino acids are not in an alpha helix or beta sheet conformation. The close approach can be defined as carbon alpha distance between two amino acids being smaller than 7.0 Å. Such two amino acids may or may not have hydrogen bond between them. A loop is far less ordered compared to alpha helix or beta sheet [5].

Even though loops are said to be less ordered than helices or sheets, some of the loop regions are relatively more ordered than others. They are called turns and their general purpose is to change the direction of a poly-peptide chain. Turns are grouped by the number of amino acids that makes up the turn and the dihedral angles, the number most commonly being three or four. The most common type of turn is β turn, which consists of four amino acids, and the first and fourth amino acids form a hydrogen bond. The γ turn contains three amino acids, the hydrogen bond is between the first and third residue. γ turn serves the purpose of linking two antiparallel beta sheets. Other types of turns exist, depending on the dihedral angles and number of amino acids they contain. [?]

The tertiary structure describes a protein's shape, or technically referred as its *fold*. It is defined as the spatial arrangement of atoms that make up the protein. The tertiary structure can be given as a set of three dimensional coordinates (x,y,z) where each coordinate corresponds to one atom of the protein. Tertiary structure is largely determined by the primary structure. The problem of predicting the tertiary structure of a protein from its primary structure is known as the protein folding problem and is one the major problems of computational biology.

Most proteins are made up from multiple poly-peptide chains, referred as subunits. The interaction and arrangement of the subunits in a protein make up the quaternary structure of that protein. The interfaces between subunits allow for formation of binding sites which monomeric proteins (single chain proteins) cannot attain. Quaternary structure or organization of subunits to create complex proteins increases functional versatility of proteins.

There can be actually one more level of structure additional to the four levels mentioned before. However this level is not actually recognized like the ones mentioned before but it's more like a structural organization unit. This unit is called structural domain or just domain for short.

2.4 Domains

During the folding process, the tertiary structure may organize around more than one structural unit. Such structural units are called domains. One subunit may consist of one or more than one domain. An absolute definition of domain does not exist but a widely accepted one is that domains are autonomous folding units within a protein. [6]

Domains are important because they have certain duties in fulfilling the biological duties of the protein they are a part of. Proteins from different organisms that have functionally similar duties share domains. Since domains are functional units and structure determines function, each domain has its specific shape, or fold and can be classified on this basis. Three major classification schemes exist. There is the α class that contains domains consisting mainly of alpha helices, β class containing domains consisting mainly of beta sheets and α + β which contains both of the elements. [7]

One example of a domain is the calcium binding domain, which has the duty of binding calcium atoms as its name implies. The calcium binding process can be observed in different parts of the body for different purposes, e.g. muscles and bones. In muscle cells calcium binding occurs during contraction, whereas calcium is used in the bones to reinforce structure. Proteins from muscles and bones, involved in these processes, contain the calcium binding domain even though their main purpose may be different. The calcium domain has a specific EF hand motif that is formed of two helices connected by a loop region. To bind the calcium atom, specific amino acids must exist in the helix and the loop region, like the presence of aspartic acids or glutamic acids in the loop region.

Another major domain type is the DNA-binding domain. These domains are included in proteins that bind to DNA for different purposes like gene expression or DNA packing. Different types of DNA binding domains can be observed. One example is zinc finger domain, named so because it contains zinc atoms. This domain contains four amino acids at specific locations which may be histidines or cysteines, which act as binding residues for

zinc atoms. Proteins that contain the zinc finger domain are generally involved in transcription regulation duties.

As stated before, the domains are functional units of the protein and make one realize the variety of functional properties of a protein.

2.5 Protein Function

The functional duties of proteins can be very diverse. Almost all functional duties in the biochemical and biophysical processes in an organism are fulfilled or assisted by proteins. Some of the proteins are enzymes that catalyze biochemical reactions. Others may fulfill mechanical functions like binding certain metal atoms for specific tasks like transportation or biochemical reactions. Proteins may also bind other proteins for in cell signaling; to form protein complexes or one protein may modify another. Moreover proteins interact with other biochemical compounds like RNA, DNA and ATP to facilitate the processes these compounds are involved in.

The diverse functionality of proteins is dependent on their ability to bind to other molecules selectively. The region that binds is called the binding site or pocket. The binding process is dependent on the structure and amino acid composition of the protein. The shape of the pocket and the physiochemical properties of amino acids in the pocket allow for selective binding. In short, protein function is directly dictated by protein structure.

2.6 Protein Structure Determination and Folding Problem

Determining the correct structure of a protein is calculating the exact three dimensional coordinates of the atoms (excluding hydrogen atoms) that make up the protein. Two widely accepted experimental methods are X-ray crystallography and NMR spectroscopy. X-ray crystallography calculates atom coordinates by measurements done on the diffraction patterns of X-rays through protein crystals [8]. NMR does its calculations

based on multidimensional nuclear magnetic resonance experiments on purified samples of aqueous proteins [9].

In the X-ray crystallography methods, X-ray beams are directed to a protein crystal, causing the electrons of the crystal to emit x-rays. By making calculations on the x-rays emitted by the electrons, the electron density of the molecule is calculated, which facilitates the building of a molecular model. This method requires the crystallization of the protein, a long and difficult process, thus cannot be used to discover the structures of proteins that cannot be crystallized, such as membrane proteins. [10]

Certain atoms have magnetic moment (spin) properties that allow them to react to RF pulses when they are aligned in a strong magnetic field. NMR method makes use of this property, by making calculations on the RF radiation emission of the atoms when they are excited by RF waves, the surrounding molecular environment of each atom can be determined. By setting a set of distance constraints, possible atomic models of the molecule can be derived by using the data obtained. The samples used in NMR experiments are protein solutions, which is easier to obtain than protein crystals. However this method can be used for small and medium size proteins because signals emitted by large molecules may overlap, thus making it impossible to discern between different molecules. [11]

It must be noted that both of these methods are expensive and time consuming. However protein sequence determination is comparatively an easier and less time consuming process, thus the number of proteins with known sequences are surpassing the number protein whose structures are also known. Since the tertiary structure is mostly dependent on primary structure, a method to correctly predict the tertiary structure from primary structure is highly desirable. The search for such a method is referred as protein folding problem.

Even though the primary structure is main determinant for tertiary structure, the folding problem is far from easy because the search space is very large, the physical factors that make a protein stable are not fully understood and for some proteins, there are

secondary factors like chaperones that affect the folding process. To solve the folding problem, three kinds of approaches exist: ab initio approach that tries to predict the structure from primary structure data and physical principles, comparative protein modeling approach that tries to find the correct structure by using known structures of homologous proteins and threading approach that compares target protein with different proteins families using a function that evaluates the goodness of fit based on energy, interactions, etc..

The protein folding problem hasn't been solved yet, new approaches are being proposed. CASP (Critical Assessment of Techniques for Protein Structure Prediction) is a community experiment aiming to assess the accuracy of existing prediction methods, done every two year [12]. Unfortunately, the problem has not been solved satisfactorily, thus the only known structures are determined by conventional NMR and crystallography.

Proteins with known structures are deposited to the PDB database. PDB database contains the structure information of almost forty thousand proteins and is the universal database of proteins. The data is stored in PDB's own file format.[11]

2.7 PDB files

A PDB file stores structural information of a protein. The primary, secondary, tertiary structure information is available. If the protein is composed of more than one poly-peptide chain, information of each chain is stored with a different one letter code. Depending on the file, there may be more information about the protein or the about structure determination process of the protein, e.g. notes by authors, etc… The primary structure is stored using three letter codes of amino acids. The secondary structure is stored as a pair of numbers, indexes of first and last residue of each secondary structure element. The tertiary structure is stored as a set of three dimensional cartesian coordinates for each atom of the protein [13].

In this rest of this section, the basic idea of alignment is explained. Also existing alignment methods comparable to our method are presented.

2.8 Protein Alignment

Identification of similar subsequences or global similarities among proteins is a very critical process for computational and theoretical biological studies. The identification process is generally referred as alignment because sequences are aligned against each other where similar regions are in correspondence.

Protein alignments are useful for gaining information about newly discovered proteins. Aligning a new protein against known protein families can give clues about the new protein, as alignment scores may help us decide which protein family the new protein belongs. An alignment between two proteins can also be considered as a distance measure between those two proteins so alignments can be used for classification as well [15]. Multiple alignments can align more than two proteins and can be used to discover similar conserved regions in protein families [16]. Generally speaking, alignment process is integral part of almost all protein analysis related tasks.

Different parameters can be used in deciding homologous subsequences. Amino acids themselves are the most simple and obvious choice for evaluating similarity. However to be able to do this, the similarity between different types of amino acids have to be defined or evaluated. For this purpose, two dimensional matrices of integers, called similarity matrices, are used. Each column and row represents one amino acid and the number at the intersection of one column to another row is the similarity of the column's amino acid to row's amino acid.

Two widely used examples of similarity scoring matrices are PAM and BLOSUM. PAM (Point Accepted Mutations) matrix is built with the assumption of a preset mutation rate and by observing the mutation rates between proteins with the use of markov chains. Probability transition matrix is calculated for sequences that are one generation apart from

each other. Similarity score is calculated by finding log likelihood ratio of these probabilities over random occurrence probabilities to obtain the PAM matrix. [17].

BLOSUM (BLOcks SUbstitution matrix) is built by using the BLOCKS database which contains ungapped alignments of highly conserved regions of proteins. The substitution frequency of each amino acid to other types is counted over the database. The similarity scores of two amino acids are obtained by finding the log score of the ratio between the probabilities of actual substitution probability by random substitution probability for two amino acids. BLOSUM-N means the matrix is built by using sequences of N percent or greater sequence identity. BLOSUM-62 is the most versatile and widely used one [18].

Structural information of a protein can also be used as a parameter for alignment processes. Amino acid composition itself is not always sufficient to decide whether two subsequences are homologous or not because some very distant proteins (remote homologs) have very low sequence identity [19]. Besides most amino acids don't have unique properties thus it's possible to replace amino acids with other types for most cases. Structural information, which is a direct determinant of function, can be more sensitive in capturing similarities where sequence information fails.

2.9 Sequence Alignment Methods

Early alignment algorithm use sequence similarities to discover homologous regions. It's because of the fact that discovering amino acid sequence of a protein is a relatively easy task compared to discovering the 3-D atomic structure. In most cases, functionally similar and evolutionally close proteins have high sequence identity and for such proteins sequence alignment methods are reliable and accurate.

Sequence alignment methods try to find the maximum length of homologous subsequences among different proteins. If the subsequences were to be continuous, it'd be

an easy task to find and evaluate homologous subsequences. However during evolution, genes undergo insertion and deletion, and this obviously affects proteins. Thus any method that has to find correct alignments must address the problem of insertions and deletions [20].

Insertion and deletion lead to excess regions between optimally alignable homologous regions. Consider the alignment process of two proteins; if one of the proteins has an insert region and the other one does not, or one protein has a specific region and the other protein has that region deleted, such regions must be aligned with gap regions so that the homologous regions can be correctly aligned globally.

There are two types of sequence alignment methods. The first type is the dynamic programming based approach. Such methods build sub-solutions iteratively; new solutions are built on previous ones. The solutions are overlapping, meaning that they can be combined. By combining the sub-solutions, the global optimum solution is found. In the case of proteins, sub-solutions are short alignments of similar regions. These regions are connected by gaps, and their combination yields one final alignment.

Needleman-Wunsch algorithm and its variation, Smith Waterman algorithm are two good examples of dynamic programming alignment methods [21],[20]. These methods rigorously try to find all local non-overlapping similar segments and combine them in a single alignment, putting gaps as necessary. When aligning large proteins, these methods may have long running times. However sequence alignments are measures of similarity and can be used to compare one protein with databases of proteins to discover information. To facilitate such searches, faster algorithms are required [15].

The second type of sequence alignment methods, heuristic based approaches are used to develop fast algorithms to fulfill the need for fast database searching. Such methods lack clear biological definitions of similarity, like minimal number of mutations between sequences, but have proven to be useful in discovering relationship between proteins. They

make use of good heuristics to discover similarities. A widely used heuristic is to use k-letter words, short subsequences.

BLAST is a typical example of a heuristic based approach using words. It operates by finding most similar words between two proteins and extending the alignment from such words [15]. FASTA is another significant heuristic based algorithm. FASTA does a preliminary search to find identical segments, then optimizes the final alignment by combining the identical segments by a dynamic programming approach [22]. These methods do not guarantee to find the global optimum, both execute faster than dynamic programming based methods.

Multiple sequence alignment methods exist which align more than two proteins. Multiple alignment algorithms also make use of dynamic programming. The main concern of a multiple sequence alignment is to decide the order of alignments and their integration afterwards. The dynamic programming methods and heuristic based methods are two mainstream approaches to protein sequence alignment problem. CLUSTALW is a widely used method of multiple sequence alignment [16].

Sequence alignment algorithms are useful for discovering conserved regions among families of proteins, phylogenetic tree construction, and classification of proteins. However they have their limitations when aligning certain types of proteins, the remote homologs. Remote homologs are homologous proteins that have lower than %25 sequence similarity. It is naturally hard to discover the homologies between remote homologs using sequence alignment methods. However as with other homologous proteins, remote homologs have conserved functions [23], [24]. To discover similarities between homologous proteins with low sequence similarity, structure information -which captures functional information- should be used.

2.10 Structural Alignment Methods

Structural alignment methods try to align proteins based on their three dimensional coordinates. The resulting alignment is a superposition of amino acids where structurally similar regions are superposed on each other. RMSD (Root mean square distance) measure is one of the most universal measures to measure the goodness of a superposition which calculates the mean distance and the similarity of two structures. RMSD measure is the average distance between the C_α atoms of two aligned and superimposed proteins [25].

Structural alignments use three dimensional coordinates to align proteins, so naturally it's only applicable for proteins with known structures. As the number of proteins in PDB has increased dramatically during the recent years, structure alignment has become applicable for more and more proteins.

There are different approaches for solving the structure alignment problem but they are roughly classified into two categories, superposition and clustering [26]. Superposition methods translate and rotate one protein in three dimensional space to minimize the protein's intermolecular distance to other protein. Clustering methods cluster the amino acids and compare the intra molecular amino acid to amino acid distances of one protein to another. Our method is a clustering method as well.

2.10.1 Superposition methods

As stated before, these methods operate by rotating one protein while the other one is stable, and the alignment they discover is the one that yields the optimum superposition, measured by RMSD [27]. In this section, we present two examples of superposition methods.

MinRMS method tries to find the alignment that yield the most optimum superposition, measured by RMSD. An intermediary similarity score is not used to evaluate the alignments; instead the algorithm uses sum squared distance between C_α atoms which

can be used to calculate RMSD. Since RMSD is the target, a one amino acid residue long alignment is always the best solution. To address this problem, minRMS allows the user the set the length of the alignment, and alignments of that length are generated [28].

The alignments are discovered using a dynamic programming approach similar to Needleman-Wunsch algorithm, but MinRMS iterates over three dimensions, two dimensions on each protein's residues and one dimension over the length of the alignment. This matrix is like a pyramid, third dimension getting smaller as the length of the alignment is increases. This method is capable of lots of structural alignments with optimum RMSD values for one protein pair, however run times are on the order of a few minutes, thus it's not feasible to use in large scale searches or classification purposes.

Another example of superposition approach is presented by Taylor, based on double dynamic programming. This method centers two structures on a pair of amino acids, one from each protein, then orients the structures based on local features to achieve a superposition between two proteins. In the obtained superposition, all relationship between pairs of atoms are quantified and an alignment is obtained by using dynamic programming. All feasible and favorable pairings are tried, each yielding a superposition and a second dynamic programming step extracts the best alignment from these set of superpositions [24].

2.10.2 Clustering methods

These methods make use of the euclidian distances between atoms to find the best structure alignment. As the aim of the structure alignment is to produce the best alignment with ideal superposition among proteins, using the intra atomic distance is a direct approach to the problem. One amino acid is composed of more than one atom and some or all of the atoms may be used in distance calculations. Calculating and using all of the distance information can result in very costly algorithms so heuristics and constraints are integrated into algorithms to achieve feasible algorithms with good results.

2.10.2.1 CE

CE (Combinatorial Extension) is a widely used structure alignment method that uses inter residue distances [29]. Protein sequence is broken into and represented by a set of AFPs (aligned fragment pair). AFPs are of fixed size, it's reported that 8 is optimum size in terms of speed and accuracy. The alignment of two proteins A and B is defined as a path of AFPs in a similarity matrix S of size $(n_A\text{-}m) * (n_B\text{-}m)$ where m is the AFP size and n_A and n_B are the length of proteins.

An alignment may start from any AFP and after that consecutive AFPs are added in such an order that the next added AFP cannot contain any residue that was included in the previous AFP. Gaps are allowed but there is an upper limit to the length of a gap segment to reduce running times, the limit is 30. In the process of addition of new AFPs, not all possibilities are explored; heuristics are employed to reduce search space.

CE uses three distance measures to evaluate similarity and AFP path extension decisions. The first measure is the average of the sum of distances between residues of two different AFPs where each residue participates once. First measure is used to decide how well two AFPs combine; it's the path extension heuristic. The second measure is similar to first one but all possible distances between non-neighbor residues are averaged for two different AFPs. Second measure evaluates the goodness of a single AFP, whether two protein fragments match well. The third measure is the root mean square distance from superimposed structures and is used in the final steps to pick best alignments and optimization.

The path extension process may start at any starting point that satisfy similarity criteria in matrix S but only the longest path is kept during path extension so result is a single alignment. Three heuristics are employed in the extension process. For candidate AFP decisions, intra AFP distance should be smaller than 3 Å. The best AFP is chosen by calculating the average distance of candidate AFP to all existing AFPs in path. The

termination heuristic is based on the average distance between all possible pairings of existing AFPs.

After extension ends, the statistical significance of the longest path is calculated by evaluating the probability of finding an alignment of same length with same or less gaps or distance from a random comparison of structures from a non-redundant set. This statistical process yields a z-score. There may be further optimization if z-score is higher than a certain threshold. The z-score is also a distance measure of two proteins, the higher the score is, more similar the proteins are.

CE was also used to detect family members of a protein among a set of proteins. By aligning a probe protein with all members of a random set, CE is able to discern the proteins of same family of probe. CE can also be used to recognize protein fold, by using a set of probes. As these results show, structural alignments are useful for classification purposes as well.

2.10.2.2 DALI

DALI (Distance matrix ALIgnment) is another clustering type structural alignment method [30]. DALI makes use of residue to residue distance matrices where each residue is represented by its C_α atom. The algorithm discovers similar regions based on the idea that similar structures should have similar inter-residue distances.

After each protein is converted to a inter residue matrix, the matrices are decomposed into submatrices, in practice the algorithm uses a hexapeptide to hexapeptide submatrix. By comparing the submatrices of two proteins to each other, using a similarity metric based on the difference of inter residue distances, and structurally similar regions are found in the form of alignments. At the second step, the alignments are combined by a score maximization procedure to achieve the best alignment.

DALI is also used for classification purposes. The FSSP (Fold classification based on Structure-Structure alignment of Proteins) makes use of DALI alignments to make a classification of PDB [19]. The classification is based on exhaustive 3D comparison of all structures among remote (<%30 percent sequence identity) and medium (%30 to %70 sequence identity) using tree representation of hierarchical clusters.

As the examples of CE and DALI demonstrate, structural alignments are used to classify remotely homologous proteins where sequence alignment may prove impotent. They can also be used in fold recognition, as exemplified by FSSP.

2.10.2.3 FAST

FAST is another clustering method [27]. As with the other examples of clustering approach, FAST makes use of intra molecular distances. However FAST has a novel approach of modeling the alignment as a graph and eliminates incompatible residue pairs to reduce the computational complexity of the problem.

FAST models the alignment of two proteins as a graph, a set points and lines connecting the points. In the graph FAST uses, the points(vertices) represent a match between one amino acid from first protein and one amino acid from second, and the lines(edges) are drawn if intra molecular distances between amino acids is smaller than a threshold value. Finding the maximum clique, a subgraph where each vertex is connected to all others by an edge, of this subgraph yields the alignment between two proteins.

The algorithm is composed of four steps. Since the graph this method proposes to use can be very huge even for medium sized proteins (10^5 vertices), FAST firstly eliminates pairs (vertices) that doesn't fit local structure comparison criteria. This local structure comparison is dependent on a similarity score which is in turn dependent on the C_α distances between members of five amino acid segments. After this elimination phase, the second step of the algorithm commences, where the edges between pairs are weighed

according to the agreement of distances, angles between these two pairs of amino acids and the angles of side chain groups of the two pairs.

The third and fourth steps of the alignment build the alignment between two proteins. In the third phase, isolated pairs, or pairs connected with low scoring edges are eliminated as well to further reduce the complexity and make the global optimum stand out. Afterwards a simple dynamic programming algorithm is run on the remaining pairs to discover the best scoring alignment where weights on edges are used as similarity scores. The final step of this algorithm tries to further improve the algorithm by trying to add pre-eliminated pairs or eliminate some bad pairs from the existing alignment by dynamic programming.

FAST, as the name implies, takes shorter time to run compared to other clustering methods like CE and DALI. FAST is also similar to the method we propose because we propose to use graph theory as well, however while FAST uses graphs to model the alignment in term of matches between amino acids, we use graphs to model proteins in terms amino acids.

Structural alignment are useful and have a wider scope of use than sequence alignment methods because of their ability to capture structural similarities which may not be captured by sequence similarity, however this comes at the expense of algorithmic complexity. Furthermore, structural alignments are used for fold classification [31][32], protein structure modeling [33] and structure based function annotation [34][35] since these tasks are done using structural comparisons.

2.11 Alignment Based Domain Recognition Methods

The two widely used functional protein classification databases are SCOP and CATH. These databases classify the proteins on the basis of their fold, functional similarity and the domains they have. Such a classification is essential and aids computational and experimental studies. Discovery and listing of domains aids protein engineering studies,

whereas a comprehensive classification is essential for systematic studies of taxonomy and evolution [31][32].

However SCOP and CATH are built by human experts. As the number of solved structures increase, the need for an automated method becomes more apparent. Different methods of proteins automated classification and domain recognition exist. DALI and CE are used to build all to all protein fold classification databases.

For domain discovery and recognition, sequence alignments methods are used extensively. Most of the domains have conserved sequences, thus sequence alignments are useful for discovering structural domains existence and boundaries by analyzing multiple sequence alignments of protein families. ProDom is an example of domain recognition method that use multiple sequence alignment methods [36].

2.11.1 ProDom

The ProDom was generated from the SWISS-PROT database [37] by automated sequence comparisons to study domain arrangements within known families or new proteins. In this approach the domains are selected according to the sequence similarities between homologous domains of SWISS-PROT sequences, the comparison between the proteins is done with BLASTP to obtain a list of homologous segment pairs. Those sets are grouped into homologous segment sets by transitive closure using the MKDOM program. In order to address the domain boundaries those sets were processed either at the ends of bona fide sequences, at the ends of tandem repeats, or where sequence shuffling is detected. After the addressing of domain regions, a multiple alignment for each family was processed with MultAlin program, and a consensus sequence is determined as the best weighted average sequence for each multiple alignment.

Since this approach addresses the domains based on conserved subsequences as found in various proteins rather than the structural conservation, such conservation does not

always infer the genuine structural domains. That is why the method may not be trusted for the domains that have the structural conservation but don't have conserved subsequences.

2.11.2 Pfam

Pfam is another protein family database that makes use of multiple sequence alignments to discover domains and families [38]. Since it uses sequence alignments, Pfam is similar to ProDom. However in addition to sequence alignments, Pfam also makes use of HMM profiles and structural data when it's available to achieve better results.

In this thesis, we'll also try to discover domain boundaries of proteins by aligning two proteins that are known to contain the same domain. We expect that the regions aligned by our algorithm to be regions belonging to domains because of the properties of the similarity function we use. Pfam and ProDom are the methods we will use to compare the results of our method when we are trying to discover domain boundaries.

3 METHODS

The algorithm we propose uses a amino acid contact network model to represent the proteins [39]. The model is used to capture the tertiary structure data information of proteins, and allows for quick and accurate assessment of structural similarities. The alignment algorithm is based on the well established dynamic programming method, and is driven by a similarity function based on primary, secondary and tertiary structures.

In our contact network model, the amino acids are nodes of the network, and the links between the nodes represent the existence of contact between the amino acids where contacts are dependent on the three dimensional distances between amino acids. By using such a model, the local structure surrounding each amino acid can be captured and compared.

The contact information of a protein can be obtained via a contact map of the protein, which is built by using the three dimensional coordinates of the atoms that make up the protein (tertiary structure) [40]. After the contact map is prepared, the network can be modeled by using graph theory. In the following sections, contact maps are explained and graph theory is explored.

3.1 Contact Map

The contact map is a two dimensional matrix that contains contact information of residues of a protein. Researchers use different criteria to decide whether two amino acids are in contact or not. A widely used and accepted criteria that we will adopt as well is

defined as follows: two residues i and k are in contact if the Euclidian distance between the C_β atoms (C_α for glycine) of the residues is smaller than or equal to 7.0 Å.

Let's say the contact map matrix is M. M matrix is filled according to a certain condition, The value at row I, column K is set to 1 if i^{th} residue of the protein is in contact with the k^{th} residue of protein, else the value is 0. In short $M_{ik} = 1$ if i^{th} residue and k^{th} residue are in contact, else $M_{ik} = 0$.

Assume that a hypothetical protein P has 6 amino acids and amino acids are enumerated from one to six. Let's say that, according to the 7.0 Å contact definition, amino acid 1 is in contact with 2 and 5, amino acid 2 is in contact with 1 and 3, amino acid 3 is in contact with 2 and 4, amino acid 4 is in contact with 3, 5 and 6, amino acid 5 is in contact with 1, 2 and 4 and amino acid 6 is in contact with 4. Then the contact map of P is as presented in Figure 3.1.

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

Figure 3.1. Contact map of P

The information stored in the contact map makes it practical to model the protein as a graph. Graphs are used to model relations between a set of objects and in our case the objects are amino acids and the relations are contacts.

3.2.Graphs

A graph is a set of points and lines connecting subset of points. The points are called nodes or vertices and lines are called edges or arcs. The nodes represent objects while vertices represent the relations. A more formal definition is given in next paragraph.

A graph G is an ordered pair G = (V , E) that satisfies three conditions:

- V is a set of vertices or nodes,
- E is a set of pairs of distinct vertices, which are called edges or arcs,
- The vertices that belong to an edge are called endpoints or end vertices of the edge.

The order of a graph is $|V|$, the size of a graph is $|E|$ and the degree of a vertex is number of other vertices it is connected by edges.

This definition is the most basic definition in graph theory, where more complex graph types like directional and attributed graphs exist. However this definition is sufficient for our content. To represent the definitions explained above, the protein P whose contact map is given in Figure 3.1. is modeled as a graph in Figure 3.2.

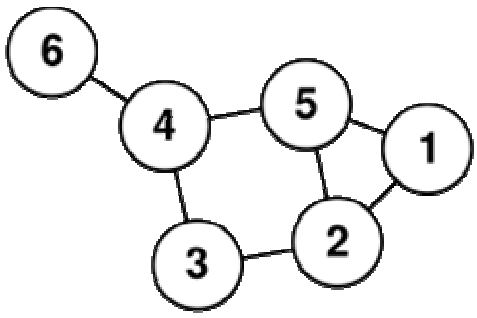

Figure 3.2. Graph of protein P

The nodes of the graph presented in Figure 3.2. represent the amino acid and labeled with numbers. The edges between the nodes signify the contacts, if two amino acid are in contact, the nodes that represent these amino acids are connected by an edge.

Two node parameter definitions will be given, which will be used in our content. But before that an additional definition must be made. Two nodes are *adjacent* if they are connected to each other by an edge. The *connectivity* of a node is the number of nodes that are adjacent to the node. This connectivity is essentially same as the node degree. The *cliquishness* of node *N* is the ratio of the number of adjacent pairs among the set of nodes adjacent to *N* to the number of all pairs in set of nodes adjacent to *N*.

Example: Connectivity of Node 4 is 3 since it has 3 adjacent nodes. Cliquishness of Node 4 is 0 / 3 because out of three possible adjacencies between Node 4 neighbors, none of them are realized. Cliquishness of Node 1 is 1 / 1 because two adjacent nodes can make 1 adjacency among themselves, and it's realized.

The contact map & graph model will be employed in our method to model tertiary structures of proteins. Structural similarities will be captured by comparing the information in graph representation.

3.3 Alignment Process

Our alignment algorithm is based on dynamic programming idea which has been widely used in computational biology, especially in alignment problems. Dynamic programming algorithms find the optimum solutions by using optimal substructures (or subsolutions). The substructures may overlap, meaning that they can be combined to yield the optimum solution. The early solutions to sequence alignment problem are two dynamic programming algorithms by Needleman-Wunsch [21] and Smith-Waterman [20] based on the observation that sequence alignment problems have overlapping substructures that can be combined.

The algorithm we present is based on Smith-Waterman algorithm, which is a variation of Needleman-Wunsch algorithm. These two algorithms are used in finding the optimum local (Smith-Waterman) and optimum global (Needleman-Wunsch) sequence

alignments. In the next section, we explain the basics of Smith Waterman algorithm since it's essential to explaining our method as well.

3.3.1 Smith-Waterman Algorithm

Let's say two protein are given as $A = a_1a_2...a_n$ and $B = b_1b_2...b_n$ and the similarity between amino acids a and b are given as s(a,b). To align two proteins of length m and n, a two dimensional matrix, H, is set first. Firstly,

$$H_{k0} = H_{0l} = 0 \text{ for } 0 \leq k \leq n \text{ and } 0 \leq l \leq m \quad (3.1)$$

The matrix is filled in such a way that H_{ij} has the maximum similarity of two segments that end in amino acids a_i and b_j, as given in the following relationship,

$$H_{ij} = \max \{H_{i-1,j-1} + s(a_i,b_j), \max_{k\geq 1}\{H_{i-k,j} - W_k\}, \max_{i\geq 1}\{H_{i,j-l} - W_i\}, 0\} \quad (3.2)$$

for $1 \leq i \leq n$ and $1 \leq j \leq m$

This relationship considers three possibilities of ending at H_{ij}, if a_i and b_j have been matched, then similarity is

$$H_{i-1,j-1} + s(a_i,b_j). \quad (3.3)$$

If the segment before a_i has been matched with k gaps, the similarity is

$$H_{i-k,j} - W_k \quad (3.4)$$

If the segment before b_j has been matched with l gaps, the similarity is

$$H_{i,j-l} - W_l \quad (3.5)$$

And 0 is included so that negative similarity cannot occur. W is a gap penalizing function, which can vary, the optimum one presented in the original article is Wk = 1 + 1 / 3 * k. The similarity function "s" can vary for DNA sequences, for protein sequences statistically derived and biologically meaningful similarity scoring matrices are used. The maximum similarity segments are found by tracking the maximum value in H. The alignment represented by the value is extracted by tracing back the values to a zero. Next best similar segment can be found by finding the second maximum value not associated with the first best similarity.

For protein sequences, Smith-Waterman algorithm is run using two dimensional similarity scoring matrices to define s function. BLOSUM [18] and PAM [19] are two examples of such matrices, statistically derived from protein sequences. These matrices are 20x20 matrices and contain the similarity score of all possible matches between 20 types of amino acids.

Such similarity scoring matrices are only capable of capturing the sequence similarities, meaning that they assign scores to matches between different types of amino acids according to specific properties of each amino acid. Matches between amino acids that exhibit similar properties are scored with positive bonuses, whereas matches between different types of amino acids are penalized with negative values.

In our method, we propose to use a combo similarity function F that replaces function s, which is built as a combination of different similarity parameters. Such a function will be capable of capturing more than just sequence similarity.

3.4 Similarity Function F

The parameters of **F** will try to capture the similarities between primary, secondary and tertiary structures of two proteins. The primary structure similarities can be captured by using a similarity scoring matrix, like most sequence alignments do. **F** uses BLOSUM62

because it's built from a sufficiently large dataset and useful for both low and high sequence identity cases.

To score the similarity of different secondary structure elements (loop, sheet, helix), a similarity scoring matrix is used [41]. The matrix is built by statistically scoring the matches between secondary structure elements in 3D_ali database [42], a database of protein alignments done by human experts. The scores are calculated by normalizing the ratio of actual match probability to random match probability for possible match combinations of secondary structure elements. The matrix is given in Table 3.1.

	H	S	L
H	2		
S	-4	4	
L	-15	-4	2

Table 3.1. Secondary structure similarity matrix

For tertiary structure comparison, we propose to model the protein as a graph based on its contact map and use cliquishness and connectivity parameters of each amino acid as a means of capturing structural (three dimensional) similarities. It's a common idea to find locally similar regions in structural alignment methods, as CE and DALI does [29][30]. In accordance with this idea, connectivity and cliquishness parameters contain local residue interaction information at two levels. Connectivity is the contact information of a residue, whereas cliquishness is the contact information of the surrounding residues. Two amino acids can be compared according to their cliquishness and connectivity values and their structural similarity can be scored.

The F function is built as a combination of the similarity scores of primary, secondary and tertiary structure. As a result of using a combination of three parameters, F function will award matches between amino acids of similar types that have similar local contact patterns and are members of same secondary structure elements.

By using the similarity function F, we aim to discover accurate alignments independent of sequence similarity. For close homologs (>%70 sequence identity), primary structure parameter is sufficient in finding the correct alignment. By adding secondary structure parameter, accurate alignments can be generated for medium (%70 to %30 sequence identity) and by adding tertiary structure along with secondary may help to determine the remote homologs (<%30 sequence identity) [19]. For medium and remote homologs, primary structure parameter can act to consolidate the matches, important matches among two cysteines or tryptophans can still have an impact while mismatches caused by non deleterious mutations will not have significant effects on the overall alignment.

The addition of tertiary structure parameter to F also has two advantages. Firstly, since we are comparing the local contacts, thus comparing local structure, our alignments can be considered structural alignments at the same time. Structural alignments are very important in many topics related to proteins and are time costly to produce. The cliquishness and connectivity parameters can be calculated relatively quickly, and their comparison cost is composed of a few arithmetic operations. Thus we obtain fast and accurate structural alignments. We obtain the structural alignments by using local alignment algorithm where gaps are allowed.

Secondly, we expect to discover domains during the alignment process, if two aligned proteins share one. Although an absolute definition of domains doesn't exist, the generally accepted definition is as follows: domains are semi-independently folding structural units, having a distinct structure aimed at fulfilling a specific function [43]. Considering this definition, It's probable that the local contact patterns of amino acids in domains should be preserved among different proteins containing the same domain. The scoring of local contact patterns make it possible to discover regions that may in fact be domains. Additionally, to capture the regions domains expected to belong to domains, we use a slightly different algorithm that captures set of alignments of short regions (generally five to twenty amino acid long). In this scheme, gaps are not allowed and mismatch penalties are higher than regular mismatch penalties.

3.5 Our Alignment Algorithm

The alignment process is based on the Smith-Waterman algorithm. To align two proteins A and B of size n and m respectively, matrix nxm M is used where M_{ij} is the maximum similarity of an alignment ending at i^{th} residue of A and k^{th} residue of B. The first initialization step is to set the first row and first column to zero.

$$M_{0j} = M_{i0} = 0 \text{ for } 0 \leq j \leq m \text{ and } 0 \leq i \leq n \tag{3.6}$$

The formula to calculate the rest of matrix M is given as:

$$M_{ij} = \max \{ M_{i-1,j-1} + F(i,k), M_{i-1,j} + G, M_{i,j-1} + G, 0 \}, \tag{3.7}$$

where F(i,k) is the similarity score of i^{th} residue of A and k^{th} residue of B and G is the gap penalty. A zero is included to make it possible to start from beginning at a point where the accumulated similarity reaches negative values. Thus the alignment is local, meaning that all of the amino acids don't have be to aligned, instead a subset of each protein's amino acid set is aligned.

The F function is the similarity function combining primary, secondary and tertiary structure similarities. If sequence (primary structure) similarity is S_{seq}, secondary structure similarity is S_{ss} and structural (tertiary structure) similarity is S_{str}, then they can be linearly combined to form F function.

$$F = k_1 . S_{seq} + k_2 . S_{str} + k_3 . S_{ss} \tag{3.8}$$

where k_1, k_2 and k_3 are the weights on each parameter. This is the general form of our similarity scoring function.

After the matrix is calculated, the maximum value is found and other matrix elements leading to maximum are found sequentially, thus the path of the alignment is discovered. Traceback ends at a zero value. The matches of amino acids, signified by diagonal direction moves, from the amino acid matches in the alignment whereas horizontal or vertical moves signify gap regions.

3.5.1 Example

A small example of the alignment process will be presented in this section. For sake of simplicity, let's say the amino acid alphabet consists of three letters: A, B and C. The sequence similarity matrix is given in Table 3.2., let it be F_{seq}. Moreover we'll use a simple structure function F_{str} for the example and secondary structure similarity will be omitted for this example. For two amino acids x and y, if their cliquishness and connectivity parameters are same, the structure similarity score is 2, else it is -2. Function F combines the sequence and structure parameters directly, k_1 and k_2 are 1.

$$F = F_{str} + F_{seq} \tag{3.9}$$

	A	**B**	**C**
A	4	-1	-1
B	-1	2	1
C	-1	1	2

Table 3.2. Similarity matrix of 3 letter amino acid alphabet

Assume that there are two hypothetical proteins P_1 and P_2 with length 4 and 6 respectively. The sequences of two proteins are: P_1 = AACC and P_2 = CAABAC.

The connectivity parameters of P_1 are (2, 3, 3, 2) and connectivity parameters of P_2 are (1, 3, 4, 3, 3, 2). The cliquishness parameters of P_1 are (1, 0.66, 0.66, 1) and cliquishness parameters of P_2 are (0, 0.33, 0.5, 0.66, 0.66, 1). Let the gap penalty be -3.

The first row and column of matrix M is set to zero, initial conditions of the algorithm according to equation (3.6). The matrix is as shown in Figure 3.3. at the beginning.

	—	C	A	A	B	A	C
—	0	0	0	0	0	0	0
A	0						
A	0						
C	0						
C	0						

Figure 3.3. Initial state of matrix M

From the initial state, some example cases of use of equation (3.7) are shown. To calculate $M_{1,1}$, we'll use equation (3.7) as follows:

$$M_{1,1} = \max\{ M_{0,0} + F(1,1), M_{0,1} + G, M_{0,1} + G, 0 \}$$
$$M_{1,1} = \max\{ 0 + (-1 + (- 2)), 0 + (-3), 0 + (-3), 0 \}$$
$$M_{1,1} = 0$$

In this calculation, the first number is the similarity score of matching 1st residue of P_1 and 1st residue of P_2. Neither structure score (-2) nor sequence score (-1) is favorable, resulting in a mismatch. Since there have not been any matches before, gapping is not sensible. Both of gap scores are negative (-3). Since matching and gapping are all negative, a zero is assigned to $M_{1,1}$. This case is an example of choosing a zero. The matrix is presented at figure 3.4. at its current state.

	—	C	A	A	B	A	C
—	0	0	0	0	0	0	0
A	0	0					
A	0						
C	0						
C	0						

Figure 3.4. Matrix M's current state

After the calculation of $M_{1,1}$, the calculation of $M_{1,2}$ is as follows:

$M_{1,2} = \max\{ M_{0,1} + F(1,2), M_{0,2} + G, M_{1,1} + G, 0 \}$

$M_{1,2} = \max\{ 0 + (4 + (-2)), 0 + (-3), 0 + (-3), 0 \}$

$M_{1,2} = 2$

In this case, the match is favorable for sequence score whereas it's unfavorable for structure score, however sequence score is higher, thus the match is a good one. Opening a gap from a previous match from $M_{0,2}$, $M_{1,1}$ is not possible since both of them are zero. This case is an example of choosing a match. The path of choice is marked with an arrow. The current state of matrix M is presented in Figure 3.5.

	—	C	A	A	B	A	C
—	0	0	0	0	0	0	0
A	0	0	2				
A	0						
C	0						
C	0						

Figure 3.5. Current state of matrix M

For the next example, the calculations until $M_{3,5}$ are not presented. After the matrix is prepared until that point, the calculation of $M_{3,5}$ is as follows,

$M_{3,5} = \max\{ M_{2,4} + F(3,6), M_{2,5} + G, M_{3,4} + G, 0 \}$

$M_{3,5} = \max\{ 3 + (-1 + 2), 6 + (-3), 7 + (-3), 0 \}$

$M_{3,5} = 4$

This case is a good example, for all possibilities can be observed here. The total similarity score of 3^{th} amino acid of P_1 and 5^{th} amino acid of P_2 is 1, a positive value, and they can be matched. However the similarity sum of previous regions may be higher. The similarity sum of regions till $M_{2,5}$ is 6 and the same sum till $M_{3,4}$ is 7. Opening a gap region from these indexes is a possibility as well. In this case, matching from $M_{2,4}$ and opening a gap from $M_{3,4}$ are equally ideal, both yielding a score of 4, thus this case can be an example of choosing a match or a gap. The matrix M at this point is shown in figure 3.6. Since it's possible to set $M_{3,5}$ from two different indexes, two arrows exist.

	—	C	A	A	B	A	C
—	0	0	0	0	0	0	0
A	0	0	2	2	0	2	0
A	0	0	2	4	3	6	0
C	0	0	0	1	7	4	
C	0						

Figure 3.6. Current state of matrix M

The final state of the matrix M is shown in figure 3.7 after all calculations are done. The maximum value is sought to find the best alignment. After it's found, the traceback procedure starts to discover the alignment that yields the found best score. The traceback procedure can simply be done by observing the path of the arrows. For this particular case, the different paths that yield the maximum score exist. In order to present the alignment that contains a gap, the path with the gap choice is preferred over the other path.

	—	C	A	A	B	A	C
—	0	0	0	0	0	0	0
A	0	0	2	2	0	2	0
A	0	0	2	4	3	6	0
C	0	0	0	1	7	4	6
C	0	0	0	0	0	4	8

Figure 3.7. Final state of matrix M

The maximum value 8 is found in $M_{4,6}$. The traceback procedure yields the path:

$M_{4,6} \rightarrow M_{3,5} \rightarrow M_{3,4} \rightarrow M_{2,3} \rightarrow M_{1,2} \rightarrow M_{0,1}$

The path starts at maximum value found and ends at a zero. All moves except $M_{3,5} \rightarrow M_{3,4}$ are matches (diagonal moves) while $M_{3,5} \rightarrow M_{3,4}$ move is a gap move. Thus the alignment is as shown as Figure 3.8.

```
P1      AAC_C
P2      AABAC
```

Figure 3.8. Resulting alignment

The fundamentals of our alignment algorithm were presented in this section. On this base, further improvements will be integrated to achieve better results and overcome the shortcomings of the presented alignment algorithm and scoring function.

One shortcoming of Smith-Waterman algorithm is the way it penalizes the gaps. When aligning two proteins, gaps are aligned against excess regions of one protein which may not be found in the other. In nature, excess regions are caused by insertions and deletions, and these deletions are insertions occur in terms of blocks. To adapt the gap penalties to this fact, affine gap penalty scheme is used [44].

3.5.2 Affine Gap Scheme

This scheme splits the gap penalties into two categories, opening gaps and extension gaps. Each gap is penalized according to its places in a gap region. If the gap in question is the first gap in a gap region, it “opens” the gap region and is penalized with gap opening penalty. If the gap is not the first gap in a gap region but merely “extends” the gap region, it is penalized with a gap extension penalty which is less than the gap opening penalty. In figure 3.9., the first gap that is aligned with D is penalized with gap opening penalty while the rest of gaps aligned with C’s are penalized with gap extension penalty.

```
AAAAAAA--------EEEEEE
AAAAAAADCCCCCCCEEEEEE
```

Figure 3.9. Example alignment for affine gaps

With the affine gap scheme, the total penalty is o + (k – 1) * e for a gap region consisting of k gaps where o is the opening penalty and e is the extension penalty. With a regular static gap penalty scheme, the penalty would be g * k, making no difference if the k gaps are a continuous bulk or not. By setting a high gap opening penalty and a low extension penalty, we favor continuous gaps over isolated single gaps.

The implementation of the affine gap scheme is similar to Smith-Waterman algorithm, to distinguish between opening and extension gaps, three matrices are used instead of one. The implementation is based on the observation that there are possible cases in an alignment process, two amino acids are aligned, one amino acid from first sequence is aligned with a gap and a gap is aligned with an amino acid from second sequence.

Each case is represented by a matrix, amino acid to amino acid case is represented by matrix A, amino acid to gap case is represented by matrix B and gap to amino acid case is matrix C. Matrices are initialized in the same manner as before. The similarity of i*th* amino acid from first sequence and k*th* amino acid from second sequence is represented by p(i,k). The gap opening penalty is o and extension penalty is e, both are negative numbers. Given these definitions, the matrices are filled by the following formulas:

$$A_{ij} = \max \{ A_{i-1,j-1} + p(i,j), B_{i-1,j-1} + p(i,j), C_{i-1,j-1} + p(i,j), 0 \} \quad (3.10)$$

$$B_{ij} = \max \{ A_{i,j-1} + o , B_{i,j-1} + e, C_{i,j-1} + o, 0 \} \quad (3.11)$$

$$C_{ij} = \max \{ A_{i-1,j} + o , B_{i-1,j} + o, C_{i-1,j} + e, 0 \} \quad (3.12)$$

Valid for $0 \leq i \leq n$, $0 \leq j \leq m$, where n is the length of first sequence and m is the length of second sequence.

Since using two matrices allows us to discern between different gaps, staying in the same matrix means we are extending an existing gap region, thus the move is penalized with only e. Moving from one matrix two another means we are starting a new gap region, thus the move is penalized with o.

3.5.3 Variations of Our Method

During the development of the thesis, we designed variants of Smith-Waterman algorithm and experimented with them to achieve better alignments. We have also tried different similarity functions and different ways of combining them. In the results section, the different sets sequence and structure scoring functions, gap penalties, and weights will be presented with each result set. Moreover, the modifications to the basic algorithm, along with their reasons and their effects on results, will be discussed.

However there's a slight variation of the algorithm that will be explained in this section. The alignments can be generated in two manners; the first way is using the local alignment method (Smith-Waterman) with regular penalties, as presented in the example. We call this method local because whenever the sum of similarity falls into negative values, a zero is used instead, ensuring that alignment can start from any location. In this scheme, only the best alignment is extracted from the M matrix. By this method we obtain structural alignments and we try to maximize the length of the alignment while keeping the RMSD as low as possible.

The second way of finding the alignment is similar to local alignment method, the generation of the matrix M and extraction procedures are same. However in this scheme, the gap penalty is set to a very large value (-1000) so there can be no gaps in the alignment. In this scheme, the alignments are gapless continuous segments which are obviously shorter compared to the alignments generated by local alignment. In this scheme, a single short

alignment cannot be the solution, thus we extract the N-best solutions and combine the ones that do not overlap. This combination approach is used in some clustering based structure alignments as well. The way the alignments are combined can vary and is explained in results section.

By using the second method, we obtain local segments of very high similarity. In this context, local is used in a different meaning than local alignment. We expect such segments to be regions belonging to domains since they exhibit highly conserved local similarity for both sequence and structure. Another factor in favor of this belief is that these local segments do not contain gaps or mismatches, thus the alignments built by using the segments are "purer" alignments compared first method.

Another feature we'll mention here is the constraint we integrated into our alignment algorithm that allows elimination of paths that create undesirable alignment. This constraint forces one to one correspondence between helices and sheets. The constraint is implemented by using additional matrices to store recent memory of matches while the alignment is being built. Amino acid matches that result in the alignment of one helix or sheet to two different helices or sheets are nullified while the alignment is being built.

3.5.4 Optimization

Our method has combined different ideas and used different variations and parameters during the development. Furthermore our similarity function linearly combines three functions using weights. Different parameter combinations affect the accuracy of the results, however the abundance of variables used in the algorithm makes manual control of parameters difficult.

To automate the parameter setting processes and to discover the set of parameters that give the best results, we have optimized our method using a genetic algorithm. Genetic algorithms are (GA for short) are heuristic search algorithms that try to find an approximate

best solution to a given problem. In our case, we have used them search for the parameter set that yields optimum results.

3.5.4.1 Specifications of optimization procedure

In order to use a genetic algorithm for optimization, a genetic representation of solution, fitness function and crossover and mutation operators have to be defined. An ideal data structure for genetic representation of our parameter set is a list. A solution S is a list of parameters S = (g_o, g_e , c_{str}, c_{seq}, c_{ss}) where g_o and g_e are integers and c_{str}, c_{seq} or c_{ss} are real numbers. g_o is the gap opening penalty, g_e is the gap extension penalty, c_{str} is the structure coefficient of structure score in the similarity function (denoted as k_1), c_{seq} is the sequence coefficient of sequence score in the similarity function (denoted as k_2) and c_{ss} is the secondary structure coefficient of secondary structure score in the similarity function (denoted as k_3) of equation 3.8. For each element of list, a range will be defined dependent on the depth of search.

The list representation is ideal for mutation and crossover. It's rather similar to binary representation, but each digit has a different domain. Crossover between two lists can be defined as a switch between one randomly picked element of same type between two lists. Mutation can be implemented by shifting a parameter's value, adding a positive or negative value. The shifting value will be small compared to size of the range of a parameter.

The fitness function choice is rather obvious because we are trying to achieve good structural alignments with low RMSD values. To eliminate alignments with high RMSD values, a fitness function that assigns high fitness values to low RMSD values is required. A rather simple choice for fitness function can be $RMSD^{-1}$, however very short alignments tend to have low RMSD values, thus such a fitness function may favor very short alignment to lower RMSD.

The length of the alignment, the number of matched amino acid pairs, must be incorporated into the fitness function to find alignments with meaningful and good superposition. If we directly use the length of alignment, the fitness function can be:

$$Fitness = \frac{3 \times length}{RMSD} \tag{3.13}$$

It was decided to multiply length by three empirically, to balance the two factors. However since we are optimizing a set of proteins, some proteins may have greater impact because they may be longer, thus the length of alignment have to be normalized. This is achieved by dividing the length of the alignment by the length of the shorter of the two proteins, normalizing the length into range between 0 and 1. The final fitness function definition is as follows: if we are aligning two proteins A and B of length *n* and *m* respectively, and the length of the A and B's alignment is *l*, the RMSD value of alignment is the RMSD, then fitness function Fit is,

$$Fitness(n, m, l, RMSD) = \frac{l}{RMSD \times \min(n, m)} \tag{3.14}$$

After making the necessary definitions, optimization process was initiated. The latest version of discrete structural similarity function was used. For sequence similarity score doubled BLOSUM62 matrix is used. The modified version of secondary structure similarity matrix is used. Gap penalty, gap extension, and coefficients of each score are parameters. The search is done with 20 solutions at each generation, for 10 generations. For randomly generated parents, the following intervals were set:

$10 \leq g_o \leq 50,$
$1 \leq g_e \leq 10,$
$0 \leq c_{str} \leq 1,$
$0 \leq c_{seq} \leq 1,$
$0 \leq c_{ss} \leq 2$

With the constraint,

$$c_{str} + c_{seq} = 1 \tag{3.15}$$

And each parameter can take values that are multiples of increments for that interval to avoid a minimum distance between randomly generated parameters. The increment for gap parameters is 1, increment for sequence and structure is 0.05 and increment for secondary structure score is 0.1. To increase the depth of search, mutation rate was set to %100.

In the following results section, the alignments generated by using different parameters, the development and integration of different parameters are presented and compared to other methods. Also the detailed specification and definitions of different variations of algorithm and implementation process of constraint are also provided.

4 RESULTS & DISCUSSION

4.1 Data

During the development of this thesis, three datasets have been used. In the first ten experiments, we've used a dataset of remote homologs with high structural similarity. This dataset is presented in a paper by Capriotti et.al. Seventeen pairs of proteins have been picked from this dataset to create our own dataset of remote homologs [45].

The second dataset is chosen from the ASTRAL database [46], [47]. ASTRAL is generated from SCOP. For the second dataset, we have picked eleven proteins belonging to three different families. The proteins were picked from a subdatabase of ASTRAL; ASTRAL40 which contains families whose sequence identity is lower than %40 percent.

The third dataset contains proteins manually picked from SCOP. The proteins were randomly chosen from the same fold, they contained the same domain or variations of the same domain but proteins had different functions. Different proteins containing ATP binding, DNA binding (winged helix, zinc finger) and calcium binding (EF hand) are chosen.

4.2 Experiment 1: Basic Algorithm

For the first ten experiments, we use the first dataset obtained from dataset of Capriotti et. al. The contact cutoff distance is 7Å. The scoring function is based on tertiary structure and sequence information. To score sequence similarity, BLOSUM62 matrix is

used. BLOSUM62 scores are doubled beforehand to bring them to same range with structure scores. Sequence similarity is denoted as F_s. Structure score is based on a function that compares connectivity and cliquishness of two residues separately and sums them. The final score is the linear combination of two parameters.

Let's say the connectivity parameters of two amino acids A and B are given as c_A and c_B and cliquishness parameters are given as ς_A and ς_B respectively. To assess connectivity and cliquishness similarity, we calculate the difference divided by the arithmetic mean for both parameters,

$$S_c\,(c_A, c_B) = \frac{2.|\,c_A - c_B\,|}{(c_A + c_B)} \tag{4.1}$$

and,

$$S_\varsigma\,(\varsigma_A, \varsigma_B) = \frac{2.|\,\varsigma_A - \varsigma_B\,|}{(\varsigma_A + \varsigma_B)} \tag{4.2}$$

The matching of A and B is awarded or penalized based on the interval in which values of S_c and S_ς are. By dividing the difference by mean, small differences between larger values will yield smaller values of S_c and S_ς compared to small differences between small values. This makes it scoring more reliable because small differences between large values are more significant. The intervals of the function is given as in the Table 4.1. where F_c signifying connectivity function and F_ς signifying cliquishness function. Both function use same values thus only connectivity function is presented.

S_c intervals	F_c result
$S_c \geq 1$	-8
$1 > S_c \geq 0.875$	-4
$0.875 > S_c \geq 0.75$	-2
$0.75 > S_c \geq 0.625$	-1
$0.625 > S_c \geq 0.5$	0
$0.5 > S_c \geq 0.375$	1
$0.375 > S_c \geq 0.25$	2
$0.25 > S_c \geq 0.125$	4
$0.125 > S_c \geq 0$	8

Table 4.1. Function intervals

These three values are combined linearly to create the final similarity scoring function F. Given two amino acids A and B, their similarity is F(A,B),

$$F(A,B) = k_1 \,.\, F_c(c_A, c_B) + k_2 \,.\, F_\varsigma(\varsigma_A, \varsigma_B) + k_3 \,.\, F_s \qquad (4.3)$$

where $k_1 = 0.25$, $k_2 = 0.25$ and $k_3 = 0.5$.

Using this function, proteins are locally aligned with gap penalty $g = 4$ and the best alignment is obtained. At this point, the RMSD measure of the alignment is not calculated, the alignments are compared with the results of existing methods. We used CE for this task. The alignment of 1A0A and 1AM9 produced by our alignment and by CE alignment are presented below in Figure 4.1.

```
1A0A:A   KRESHKHAEQARRNRLAVALHELASLI_PAEWKQQNVSAAPSKATTVEAACRYIRH____L_QQNGS
1AM9:A   KRTAHNAIEKRYRSSINDKIIELKDLVVGTE_AKLNKSAV_LRKA_ID_YIRFLQHSNQKLKQENLS
                                Our alignment

1A0A:A   RESH-KHAEQARRNRLAVALHELASL-IPAEWKQQNVSAAPSKATTVEAACR---YIRHLQQNG
1AM9:A   RGEKRTAHNAIEKRYRSSINDKIIELKDLVVGTEAKL-----------NKSAVLRKAIDYIRFL
                                CE alignment
```

Figure 4.1 Alignment Comparison

This comparison showed that our method roughly aligned the homologous regions other method aligned, but there are shifts. The RMSD values are affected as a result as presented in Table 4.2 along with the RMSD values CE alignments yield for comparison. The shifts exist because the gap penalty is not ideal and gap extension is a problem. To address this problem, affine gap penalty scheme is implemented.

	Our Method		**CE**	
Protein Pair	RMSD (Å)	Length	RMSD (Å)	Length
12AS:A 1PYS:A	22.65321541	245	3.4	210
19HC:A 1NEW:_	8.462189674	63	3.1	64
1A0A:A 1AM9:A	6.735984325	57	3.5	48
1A17:_ 1E96:B	15.4651947	144	1.8	122
1A1Z:_ 1NTC:A	16.93305969	77	3.9	40
1A28:A 1LBD:_	8.084799767	213	2.8	193
1A34:A 1AUY:A	10.54008961	119	3.7	123
1A3A:A 1A6J:A	2.939950228	139	2.3	132
1A53:A 1NSJ:_	13.20775795	186	2.7	188
1A5R:_1 UBI:_	4.232666969	74	2.5	70
1A6M:_ 1ASH:_	3.159033298	133	2.0	139
1A7T:A 1SML:A	8.585944176	213	2.2	194
1A9V:_ 1EHX:A	10.70693493	80	3.9	83
1ABA:_ 1ERV:_	6.854056835	72	3.7	75
1AC5:_ 1IVY:A	6.63659668	416	2.3	378
1ACP:_ 2AF8:_	5.65189743	75	5.3	56
1AD3:A 1BPW:A	3.283203125	416	2.3	416

Table 4.2. RMSD values of experiment 1

4.3 Experiment 2: Affine Gap Penalties

In this experiment, we implemented the affine gap penalty which is explained in methods section. Affine gap scheme was used to counter the problems explained in the previous section, and can be seen by comparing the alignments in Figure 4.1.

After affine gap penalty scheme was implemented, RMSD values of the alignments have been calculated to assess the alignments produced by our method. The gap opening is -10 and extension penalty is -1.

Protein Pair	RMSD (Å)	Length
12AS:A 1PYS:A	22.62244225	254
19HC:A 1NEW:_	8.921919823	63
1A0A:A 1AM9:A	4.378242493	54
1A17:_ 1E96:B	19.23015976	145
1A1Z:_ 1NTC:A	13.94208145	70
1A28:A 1LBD:_	10.98708439	219
1A34:A 1AUY:A	10.67560101	120
1A3A:A 1A6J:A	3.642231464	137
1A53:A 1NSJ:_	11.09077358	186
1A5R:_1 UBI:_	3.719026327	76
1A6M:_ 1ASH:_	2.622909784	138
1A7T:A 1SML:A	7.151891708	215
1A9V:_ 1EHX:A	16.9732132	82
1ABA:_ 1ERV:_	7.145294666	73
1AC5:_ 1IVY:A	9.200638771	411
1ACP:_ 2AF8:_	5.35771513	75
1AD3:A 1BPW:A	4.27207756	417

Table 4.3. RMSD values of experiment 2

The results, as presented in Table 4.3, vary for each protein. For some proteins the alignments are unacceptable, having very high RMSD values in the range of 15Å to 20Å. Some alignments are acceptable, RMSD values in 4Å to 6Å range. There are also very good alignments with RMSD values in 2 Å to 4 Å range.

Different gap penalties were tried to decide if the gap penalty choice is the cause of high RMSD values. The generally accepted ideal range for gap penalties is the range of maximum value of awarding function and the values -10,-1 comply with this idea.. In Table 4.4, the results produced by setting gap opening = -5, gap extension = -1 are presented as an instance.

It can be observed that lowering the gap penalties did not improve the results at all. The protein pairs that yielded high RMSD alignment still do so. For example, 12AS:A-1PYS:A pair yields RMSD values higher than 20Å regardless of gap penalty choice. These results show that gap penalty itself is not cause of inaccurate alignments. Therefore additional information must be incorporated into the similarity function to get lower RMSD values.

Protein Pair	RMSD (Å)	Length
12AS:A 1PYS:A	23.0240612	233
19HC:A 1NEW:_	7.849016666	65
1A0A:A 1AM9:A	5.252778053	53
1A17:_ 1E96:B	15.38170719	134
1A1Z:_ 1NTC:A	16.39752007	76
1A28:A 1LBD:_	10.90044594	215
1A34:A 1AUY:A	10.28055286	112
1A3A:A 1A6J:A	2.972669363	136
1A53:A 1NSJ:_	11.66374588	181
1A5R:_1 UBI:_	4.417040348	73
1A6M:_ 1ASH:_	3.114444017	135
1A7T:A 1SML:A	10.25722122	208
1A9V:_ 1EHX:A	17.35075378	82
1ABA:_ 1ERV:_	9.444479942	66
1AC5:_ 1IVY:A	10.29088402	399
1ACP:_ 2AF8:_	5.35771513	75
1AD3:A 1BPW:A	5.102340221	408

Table 4.4. RMSD values of experiment 2 with different gap penalties

4.4 Experiment 3: Secondary Structure Similarity Matrix

At this stage, the similarity function is based on sequence and structure parameters. Sequence similarity function is essentially comparing the primary structure similarity of proteins. Structure similarity function is based on cliquishness and connectivity parameters which model tertiary structure information of proteins. To further consolidate the similarity function, secondary structure information can be incorporated into similarity function.

The secondary structure similarity score is denoted by F_{SS} and is obtained by using the secondary structure similarity matrix as explained in Methods sections. F_{SS} is added to combination of sequence and structure similarity scores. The original values in the matrix are not in the range of other similarity functions so F_{SS} function is multiplied by two. The F function changes as follows,

$$F(A,B) = k_1 \,.\, F_c(c_{A,}\, c_B) + k_2 \,.\, F_{\varsigma}\,(\varsigma_{A,}\, \varsigma_B) + k_3 \,.\, F_s + 2 \,.\, F_{SS} \qquad (4.4)$$

where $k_1 = 0.25$, $k_2 = 0.25$ and $k_3 = 0.5$.

The results produced by the new function are given in Table 4.5. with gap opening penalty = -10, gap extension penalty = -5. The secondary structure information improved the results for 12AS:A-1PYS:A and 1A34:A-1AUY:A pairs. Especially the first pair has its RMSD value halved from 20 Å to approximately 10Å. The RMSD values of some alignments haven't improved at all (1A17:_ - 1E96:B, 1A1Z:_ - 1NTC:A, 1A28:A - 1LBD:_).

Protein Pair	RMSD (Å)	Length
12AS:A 1PYS:A	9.664269447	215
19HC:A 1NEW:_	6.898514271	68
1A0A:A 1AM9:A	4.484011173	47
1A17:_ 1E96:B	16.99118996	147
1A1Z:_ 1NTC:A	11.63062382	63
1A28:A 1LBD:_	12.53302288	212
1A34:A 1AUY:A	5.549195766	116
1A3A:A 1A6J:A	3.521498442	135
1A53:A 1NSJ:_	7.775279999	178
1A5R:_1 UBI:_	4.253891468	76
1A6M:_ 1ASH:_	2.204893351	137
1A7T:A 1SML:A	5.952233791	201
1A9V:_ 1EHX:A	16.37402916	75
1ABA:_ 1ERV:_	4.778537273	67
1AC5:_ 1IVY:A	8.032186508	402
1ACP:_ 2AF8:_	5.286103249	75
1AD3:A 1BPW:A	3.368440628	408

Table 4.5. RMSD values of experiment 3

Since we have added a new parameter to F function, the maxima of F function has increased as well. Gap penalties should be changed accordingly, so higher penalties than -10,-1 has been tried to see if better results can be obtained. In Table 4.6., results with higher gap opening penalty are presented. The gap opening penalty has been increased to -15, extension penalty is same, still -1. This change hasn't done a lot of contribution but has improved almost all alignments marginally, except for 1A28:A - 1LBD:_ pair whose RMSD has improved a lot.

Protein Pair	RMSD (Å)	Length
12AS:A 1PYS:A	9.594562531	216
19HC:A 1NEW:_	6.753612041	68
1A0A:A 1AM9:A	4.605853558	48
1A17:_ 1E96:B	17.23191834	150
1A1Z:_ 1NTC:A	11.63062382	63
1A28:A 1LBD:_	7.569907665	210
1A34:A 1AUY:A	4.88739872	119
1A3A:A 1A6J:A	3.466456652	133
1A53:A 1NSJ:_	7.801670551	178
1A5R:_1 UBI:_	3.740767479	75
1A6M:_ 1ASH:_	2.204893351	137
1A7T:A 1SML:A	5.551774025	196
1A9V:_ 1EHX:A	16.15745354	78
1ABA:_ 1ERV:_	4.495889187	67
1AC5:_ 1IVY:A	7.979990005	407
1ACP:_ 2AF8:_	5.286103249	75
1AD3:A 1BPW:A	3.222174883	414

Table 4.6. RMSD values of experiment 3 with different gap penalties

4.5 Experiment 4: Secondary Structure Similarity Matrix Modified

It can be observed in the secondary similarity matrix that helix to helix similarity and loop to loop similarity is equal. However helices are more ordered secondary structure elements compared to loops. We want to emphasize helix matches over loop matches thus we made some empirical modifications to secondary structure matrix. Helix to helix match score is increased to 4 and sheet to sheet match has been increased to 6 to preserve the difference between the two scores. Furthermore, generally a sheet is shorter than a helix, so helix similarity contributions in the overall alignment can shadow the significance of the sheet similarity contributions without increasing the sheet to sheet match score. The modified secondary structure matrix is given in Table 4.7.

	H	S	L
H	4		
S	-4	6	
L	-15	-4	2

Table 4.7. Modified secondary similarity matrix

Results with the modified secondary structure are presented in Table 4.8. Different gap penalties have been tried since the change in secondary structure similarity matrix changes affects the maxima of F. For results presented in Table 4.8.: gap opening = -20,gap extension = -1. Thus the gap opening penalty has been increased again according to the increase in F function.

Protein Pair	RMSD (Å)	Length
12AS:A 1PYS:A	8.438858032	219
19HC:A 1NEW:_	6.753612041	68
1A0A:A 1AM9:A	4.605853558	48
1A17:_ 1E96:B	17.23191834	150
1A1Z:_ 1NTC:A	11.63062382	63
1A28:A 1LBD:_	7.153425694	210
1A34:A 1AUY:A	5.078693867	126
1A3A:A 1A6J:A	3.466456652	133
1A53:A 1NSJ:_	7.688687801	182
1A5R:_1 UBI:_	3.740767479	75
1A6M:_ 1ASH:_	2.204893351	137
1A7T:A 1SML:A	5.537323952	198
1A9V:_ 1EHX:A	15.80622292	77
1ABA:_ 1ERV:_	4.37745285	68
1AC5:_ 1IVY:A	5.582899094	402
1ACP:_ 2AF8:_	5.286103249	75
1AD3:A 1BPW:A	3.21536088	415

Table 4.8. RMSD values of experiment 4

The results in Table 4.8. show marginal improvements, except for 1AC5:_ - 1IVY:A pair. For this pair, the RMSD improvement is approximately 3Å, which brings the overall

RMSD to 5 Å, into the range of acceptable RMSD interval (2 Å to 5Å) for protein alignments. Even though the change in secondary similarity matrix does not contribute much at this step, there is improvement and for the reasons stated before, the changes to matrix are kept for the remaining experiments.

4.6 Experiment 5: Continuous Structure Function

At this point, results have been improved significantly for some proteins. Affine gap penalty scheme and secondary structure information has reduced RMSD values for almost all protein pairs with two exceptions. For some alignments, the improvements have been drastic, the change was more than 10Å as in the case of 12AS:A 1PYS:A pair. In the beginning, some alignments' RMSD values unacceptable (more than 10Å) and their RMSD values have been reduced to acceptable values (less than 5Å). For 1A17:_ - 1E96:B, 1A1Z:_ - 1NTC:A and 1A9V:_ - 1EHX:A pairs, alignments still have unacceptable RMSD values, larger than 10 Å. Thus we conclude that affine gap scheme and secondary structure information are not sufficient additions to our method to extract acceptable alignments from these three pairs. Rather than adding new parameters to current similarity function, existing parameters are to be optimized.

Considering the current parameters, the structure similarity function, which is the combination of cliquishness and connectivity similarity functions, is assessing the similarity roughly. The award is dependent on the ratio of difference to average for cliquishness and connectivity parameters. This ratio can be misleading in certain cases. For example, assume that two amino acids A,B have connectivity values 1 and 2 respectively and two amino acids C,D have connectivity values 4 and 8. Since the ratio of difference to average is equal for pairs (1,2) and (4,8), thus

$$F_c(c_A, c_B) = F_c(c_C, c_D) \tag{4.5}$$

The same case is true for cliquishness function F_ς as well. This property is obviously undesired because the similarity is being misjudged. According to Atılgan et. al. on a study

of 54 proteins at 7Å contact distance, connectivity parameter of proteins are in the range of 2 to 14, majority of them in 4 to 10 range. Considering this statistic, 1 and 2 are very close values and should be awarded while 4 and 8 are distant values and shouldn't be awarded at all.

To fix this problem, a different structure similarity function is designed. This function is not made of separate two functions, a cliquishness similarity function and a connectivity similarity function, like the old one but combines the both parameters directly to assess structure similarity as modeled by connectivity and cliquishness measures. The function uses the absolute difference of both connectivity and cliquishness parameters. Logarithm function is used to make the scores exponential. The structure similarity function F_S for two amino acids A and B is given as,

$$F_s(c_A, c_B, \varsigma_A, \varsigma_B) = 10.\left(1 - \left|\varsigma_A - \varsigma_B\right|\right).\left(1 + \log_2 \frac{1}{\left|c_A - c_B\right| + 1}\right) \quad (4.6)$$

where ς_A, ς_B are cliquishness values and c_A, c_B are connectivity values of amino acids A and B.

Since there is only one structure similarity function now, the linear combination of similarity function F is modified as follows

$$F(A,B) = k_1 \,.\, F_s(c_A, c_B, \varsigma_A, \varsigma_B) + k_2 \,.\, F_{seq} + 2 \,.\, F_{SS} \quad (4.7)$$

where $k_1 = 0.5$ and $k_2 = 0.5$.

The results generated by using the modified structure function are presented in Table 4.9. Different gap penalties have been tried and the presented results are with gap opening = -20, gap extension = -1. The new function has increased RMSD value of some alignments and reduced one of them. The improved alignment is of 1A53:A 1NSJ:_, and the RMSD is 4Å. Up until now, the changes introduced haven't been able bring this pair's alignment to

4Å. However the new structure function F_s has increased the average RMSD thus it cannot be considered an improvement.

Protein Pair	RMSD (Å)	Length
12AS:A 1PYS:A	8.644621849	219
19HC:A 1NEW:_	8.782649994	60
1A0A:A 1AM9:A	4.42521143	47
1A17:_ 1E96:B	17.02920151	150
1A1Z:_ 1NTC:A	11.52779865	62
1A28:A 1LBD:_	6.725256443	206
1A34:A 1AUY:A	6.482067585	106
1A3A:A 1A6J:A	3.378324032	131
1A53:A 1NSJ:_	4.740402222	173
1A5R:_1 UBI:_	3.740767479	75
1A6M:_ 1ASH:_	2.204893351	137
1A7T:A 1SML:A	5.616611958	192
1A9V:_ 1EHX:A	15.85970306	79
1ABA:_ 1ERV:_	4.927250385	65
1AC5:_ 1IVY:A	5.881895065	388
1ACP:_ 2AF8:_	6.832217693	75
1AD3:A 1BPW:A	3.530610323	403

Table 4.9. RMSD values of experiment 5

4.7 Experiment 6: Discrete Structure Function

Given that the continuous structure similarity function hasn't reduced the overall RMSD values, another structure function is designed. The new structure function proposed is discrete and the decision of penalizing or awarding is based on connectivity values and absolute differences of cliquishness and connectivity. The design of function is based on the observation that domains are well connected which means that amino acids in domains have large connectivity values. Thus structural matches in well connected regions are awarded more than other regions. The connectivity threshold to decide if an amino acid is well connected is 6.

In the process of scoring the structure similarity of two amino acids, the first criteria is to check if both amino acids' connectivity values are higher than 6. The second criteria is the absolute difference of connectivity values of amino acids. The last criteria is the absolute difference of cliquishness values. The function branches according to these criteria and assigns an award or a penalty to the match.

The function can be presented best in a flowchart. If both amino acids' connectivity values are larger than or equal to 6 and the absolute difference of connectivity measures is 1, then:

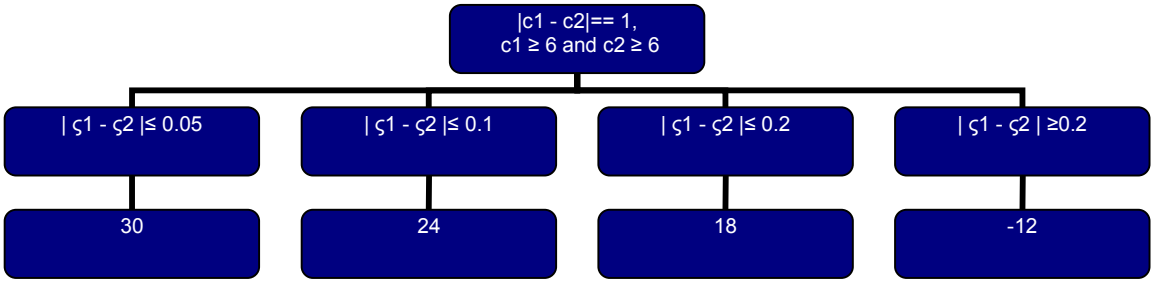

Figure 4.2. First tree of function

The score assigned by function is given in the last node. If both amino acids' connectivity values are larger than or equal to 6 and the absolute difference of connectivity measures is 2, then:

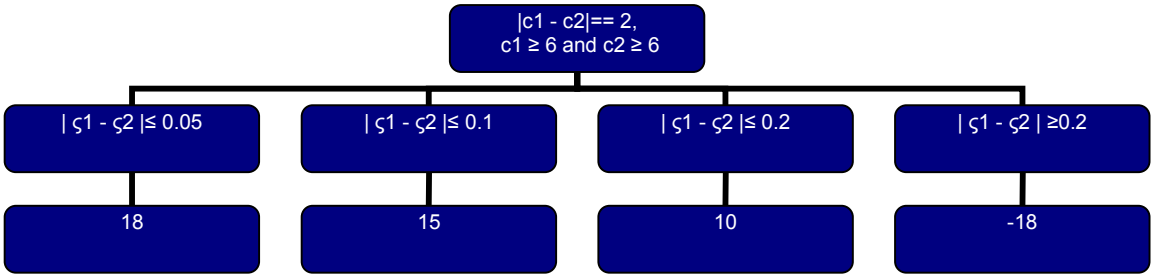

Figure 4.3. Second tree of function

These two charts show the cases for matches between well connected amino acids. If one of amino acids' connectivity measure is smaller than 6 and the absolute connectivity difference is 1, then:

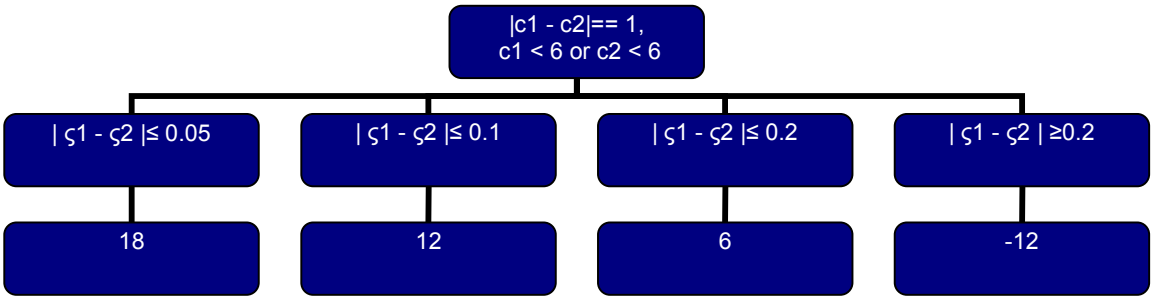

Figure 4.5. Third tree of function

If the absolute difference is 2, then

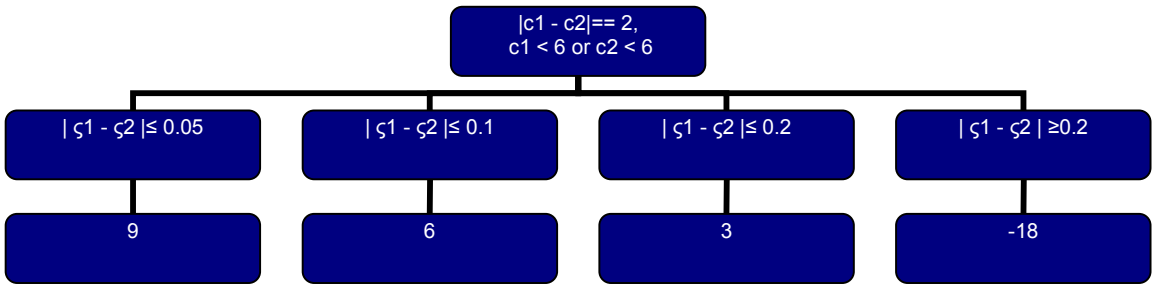

Figure 4.6. Fourth tree of function

For the remaining conditions where absolute connectivity difference is larger than 2, all matches will be penalized. The penalty is simply calculated by the formula,

$$F_s(c_A, c_B, \varsigma_A, \varsigma_B) = \left(\left|\varsigma_A - \varsigma_B\right| . \left|c_A - c_B\right| . -15\right) - 10 \qquad (4.8)$$

The results obtained by using this new function are presented in Table 4.10. After trying different gap penalties, gap opening penalty is set to -30 and extension penalty is set to -1. Gap opening penalty has been increased as the structural function maxima is higher now. Increasing the gap penalty has reduced the length of some alignments thus reducing the RMSD values marginally but some alignments have improved. RMSD values of pairs

1A17:_ - 1E96:B and 1A9V:_ - 1EHX:A have been over 15Å up until now, with the new function they have been lowered to approximately 10Å. Plus, 1A53:A - 1NSJ:_ pair's RMSD measure has been reduced to 4Å from 7Å. But RMSD values of two pairs, 19HC:A 1NEW:_ and 1A34:A 1AUY:A, has increased by approximately 4Å. New function has contributed, allowing us to reduce the RMSD values of pairs that have been unacceptable until now but needs to be optimized overall.

Protein Pair	RMSD (Å)	Length
12AS:A 1PYS:A	7.488014221	204
19HC:A 1NEW:_	10.03957939	45
1A0A:A 1AM9:A	4.277388096	47
1A17:_ 1E96:B	10.02275658	138
1A1Z:_ 1NTC:A	10.8199892	58
1A28:A 1LBD:_	7.208533287	201
1A34:A 1AUY:A	9.747236252	97
1A3A:A 1A6J:A	3.080663919	133
1A53:A 1NSJ:_	4.113963127	178
1A5R:_1 UBI:_	3.623967409	65
1A6M:_ 1ASH:_	2.722320795	137
1A7T:A 1SML:A	4.667170048	192
1A9V:_ 1EHX:A	9.643212318	74
1ABA:_ 1ERV:_	5.270978928	69
1AC5:_ 1IVY:A	4.855390072	386
1ACP:_ 2AF8:_	4.911249161	70
1AD3:A 1BPW:A	2.988206148	403

Table 4.10. RMSD values of experiment 6

4.8 Experiment 7: Discrete Structure Function Modified

The discrete structure function has a well defined award scheme but the penalties are fixed for nearly all penalty situations. We decided to fix this problem with a simple change. Our observations showed that the majority of gaps in alignments are in loop regions between concentrated sheet and helix regions. Furthermore two loop regions from two homologous proteins exhibit less structural similarity compared to helices and sheets since

loops are not as ordered as helices and sheets. The discrete structure function was modified according to these observations.

When calculating F_s for two amino acids A and B, if A or B is part of a loop, then F_s is ,

$$F_s(c_A, c_B, \varsigma_A, \varsigma_B) = \left(|\varsigma_A - \varsigma_B| . |c_A - c_B| . -15\right) - 6 \quad (4.9)$$

Else the F_s is,

$$F_s(c_A, c_B, \varsigma_A, \varsigma_B) = \left(|\varsigma_A - \varsigma_B| . |c_A - c_B| . -15\right) - 12 \quad (4.10)$$

By modifying the function this way, we hope to introduce less gaps and reduce the sensitivity of alignments to gap parameter.

The results generated with the modified F_s are presented in Table 4.11. Two sets of results are displayed, each one with a different gap penalty. The first set is generated with gap opening = -30, gap extension = -3. Second set is generated with gap opening = -40, gap extension = -4. Reducing the penalty for loop regions didn't have an impact on the results. There is a specific problem that can be observed with comparing the two sets of results. Gap parameters that reduce the RMSD of 1A17:_ - 1E96:B pair, increase the RMSD of 1A9V:_ - 1EHX:A pair, resulting in a trade off situation. This trade off is because of the fact that by using the same gap penalty for all alignments we cannot optimize all the alignments without modifying other parameters.

	Gap opening = -30, extension = -3		Gap opening = -40, extension = -4	
Protein Pair	RMSD (Å)	Length	RMSD (Å)	Length
12AS:A 1PYS:A	7.698416233	223	8.501337051	217
19HC:A 1NEW:_	6.938711643	38	6.643115044	34
1A0A:A 1AM9:A	4.265600204	46	4.265600204	46
1A17:_ 1E96:B	16.22870636	146	9.948364258	139
1A1Z:_ 1NTC:A	12.71069241	61	12.71069241	61
1A28:A 1LBD:_	7.208533287	201	6.557795525	211
1A34:A 1AUY:A	5.166750908	119	5.373473167	120
1A3A:A 1A6J:A	3.080663919	133	2.722803116	136
1A53:A 1NSJ:_	4.042504787	181	4.050766468	182
1A5R:_1 UBI:_	4.193257332	75	2.813886166	72
1A6M:_ 1ASH:_	2.722320795	137	2.722320795	137
1A7T:A 1SML:A	6.040650845	203	5.710077763	200
1A9V:_ 1EHX:A	9.213345528	80	15.00353336	75
1ABA:_ 1ERV:_	5.118096352	70	5.101519585	72
1AC5:_ 1IVY:A	5.39422226	405	5.491436005	391
1ACP:_ 2AF8:_	5.149586201	74	5.149586201	74
1AD3:A 1BPW:A	3.196627378	421	2.954343557	422

Table 4.11. RMSD values of experiment 7

4.9 Experiment 8: Strip Combining Approach

By observing the results of our experiments, we discovered that gap penalty was very important factor in determining the RMSD value of the alignments. Two alignments generated with same parameters except gap parameter can align almost exactly same regions with very small shifts, but even small shifts can affect RMSD greatly. This property of the RMSD measure can be observed in the example of the alignment of pair 1A17:_ - 1E96:B.

```
1A17:_      KRAEELKTQANDYFKAKDYENAIKFYSQAIELNP_SNAIYYGNRSLAYLR
1E96:B      VEAISLWNEGVLAADKKDWKGAL____DAFSAVQDPHSRICFNIGCMYTI

1A17:_      TECYGYALGDATRAIELDKKYIKGYYRRAASNMALGKFRAALRDYETVVK
1E96:B      LKNMTEAEKAFTRSINRDKHLAVAYFQRGMLYYQTEKYDLAIKDLKEALI

1A17:_      VKPHDK_________________DA_____________KMKYQECNKIVKQ
1E96:B      QLRGNQLIDYKILGLQFKLFACEVLYNIAFMYAKKEEWKKAEEQLALATS

1A17:_      KAFERAIAGDEHKRSVVDSLDI_ESMTIEDE
1E96:B      MKSEPRHSKIDKAMECVWKQKLYEPVVIPVG
Gap opening = -30, gap extension = -3, RMSD = 16.23

1A17:_      KRAEELKTQANDYFKAKDYENAIKFYSQAIELNPSNAIYYGNRSLAYLRT
1E96:B      VEAISLWNEGVLAADKKDWKGALDAFS___AVQDPHSRICFNIGCMYTIL

1A17:_      ECYGYALGDATRAIELDKKYIKGYYRRAASNMALGKFRAALRDYETVVKV
1E96:B      KNMTEAEKAFTRSINRDKHLAVAYFQRGMLYYQTEKYDLAIKDLKEALIQ

1A17:_      KPHDK_________________DA_____________KMKYQECNKIVKQK
1E96:B      LRGNQLIDYKILGLQFKLFACEVLYNIAFMYAKKEEWKKAEEQLALATSM

1A17:_      AFERAIAGDEHKRSVVDSLDI
1E96:B      KSEPRHSKIDKAMECVWKQKL
Gap opening = -40, gap extension = -4, RMSD = 9.95
```

Figure 4.7. Different alignments of our method

The two alignments are very similar, there is only one residue shift until the first gap opening and after a few gap regions the two alignments stay identical to end, except for an extra region the first alignment contains. Shift and the extra region at the first alignment's end results in a 5Å difference in RMSD values.

To attack this problem, we tried to exclude the gap concept out of our alignments. As it can be observed in the example alignments, alignments are generally composed of long alignments of amino acid regions without gaps (we'll call them strips) and long gap regions.

Short aligned amino acid regions and short gap regions do exist, but they are rare compared to strips. So we can think of our alignments as a set of strips connected with gap regions. However the strips are not independent of each other, the way two strips can be connected is affected by the gaps between, thus the gap penalty parameter itself. If the strips making up the alignment can be obtained one by one and combined to obtain the correct alignment, the gap penalty won't be a parameter anymore.

To implement the idea explained in the previous paragraph, proteins are aligned using the same similarity function, but the gap opening parameters is set to a sufficiently large number, e.g. -1000, to prevent generating alignments containing gaps. After the alignments matrix M is prepared, not only the 1st best alignment is extracted, but the 50 best alignments are extracted. As the gap penalty is very high, we expect these alignments to represent strips. The strips will be combined to obtain the correct alignment we want to find. However some of the strips may overlap, thus the combination process is not a straightforward one.

We want to construct a set of non-overlapping strips to form a long and accurate alignment. Adding one strip at a time and checking if the added strip overlaps with existing ones is an exhaustive approach and bound to consume a lot of time. To hasten the combination process, overlapping strips are collected in buckets, a set of strips in which each strip overlaps with at least another strip. After the strips are placed in buckets, sets of strips can be generated by picking one strip from each bucket and adding each to a strip set. Each set of strips represents a different alignment, generating multiple solutions. This makes sense, as fifty strips can be combined in different combinations.

Some of the handpicked results are presented in Table 4.12. We picked some of the longest results and some of lowest RMSD alignments. Since this new approach was meant to solve pairs with poor results, we worked on 4 specific pairs. 12AS:A - 1PYS:A pair has found the correct alignment, but there was no improvement. For 1A28:A – 1LBD:_ pair, even the lowest RMSD attained is higher than the earlier results. The same case holds for 1A9V:_ - 1EHX:A pair, the new results are worse. 1AC5:_ - 1IVY:A pair found correct

alignment like earlier results but there is no improvement in terms of RMSD and the length of the alignment is shorter.

Protein Pair	RMSD (Å)	Length
12AS:A 1PYS:A	7.806663513	216
1A28:A 1LBD:_	7.957023144	43
1A28:A 1LBD:_	17.57217598	134
1A9V:_ 1EHX:A	6.869467258	37
1A9V:_ 1EHX:A	3.313695669	11
1AC5:_ 1IVY:A	2.97833848	110
1AC5:_ 1IVY:A	5.939013958	316

Table 4.12. RMSD values of experiment 8

The strip combining approach's limitations lie in the fact that for some pairs, the strips extracted overlap too much, so too many fall into the same bucket, limiting the search space. However if more strips are extracted, runtime will increase too much, limiting the feasibility of the algorithm. Furthermore for 12AS:A – 1PYS:A pair, more than a thousand solutions are generated, however the result again converged approximately to the limits we attained with earlier results.

4.10 Experiment 9: 1-1 Correspondence Constraint.

Seeing that strip combining approach did not improve our results, we again concentrated on our previous approach. Gap penalties are again included in our parameter set. To understand why our results are unsatisfactory for some proteins, we analyzed the alignments generated by our method in detail. We also did comparisons with alignments generated by existing methods.

During the analysis, we observed that one helix or sheet from one sequence can be aligned with two or possibly more helices or sheets respectively. This situation is not desirable, one to one correspondence between helices and sheets of homologous proteins is expected because each of these secondary structure elements in a protein is an element of the fold the protein is part of.

The problem can be observed in Figure 4.7. A small subset of the alignment of two proteins is presented. Let's denote the first protein as P_A and second as P_B. The protein sequences are in the first two lines and the secondary structure types of each amino in the sequence are presented in the following lines. In this short alignment, the loop between two helices of P_B is aligned with gaps, thus the single helix of P_A is aligned with two helices of P_B.

P_A sequence	V	T	D	_	_	_	_	_	S	L
P_B sequence	M	K	A	R	G	V	T	P	R	D
P_A secondary structures	H	H	H	_	_	_	_	_	H	H
P_B secondary structures	H	H	H	L	L	L	L	H	H	H

Figure 4.8. Alignment of one helix to two helices

We have included a constraint in our algorithm to force one to one correspondence among helices and sheets. The basic idea is make the matrix setting process intelligent. When scoring a match between two amino acids, the secondary structure types of the previous amino acid match are checked. If this match violates one to one correspondence property, the match is scored with a large negative integer, nullifying it. However the previous match information is not always readily available, because of the presence of gaps in the alignment. As exemplified in Figure 4.7. the match before S-R is A-D, however this information is not directly accessible because there are gaps between the pairs. We also cannot decide the matched pair before the current pair by looking at the alignment because the alignment is not fully determined while it's being calculated. To circumvent this difficulty, the memory of recent matches has to be kept.

Three match information memory matrices are introduced to store the secondary structure information of amino acid pairs that have been matched. There are three memory matrices because affine gap penalty scheme uses three matrices. The memory matrices are D,E,F keeping the matched pair information of A,B,C respectively.

The conditions are different for D,E and F because D matrix stores the match memory of A matrix which is the matrix of amino acid to amino acid matches. E and F matrices store the memory of B and C matrices which in turn store the memory of gap to amino acid matches.

For D matrix, the rules are as follows: to set D(i,k), the if the new matched pair (i^{th} amino acid and k^{th} amino acid) is a match of same types of secondary structures, no memory is necessary and the new match is stored in the matrix. If one member of the amino acid pair (i^{th} amino acid or k^{th} amino acid) is in a loop, the previous match - whether it's in D(i-1,k-1), E(i-1,k-1) or F(i-1,k-1) - is stored in to D(i,k). The reasoning of this rule can be observed in Figure 4.8. The match between R-R is a match between a helix and loop, thus the first helix of P_B is now out of picture. The last match has to be remembered so that the helix of P_A is not matched to another helix of P_B.

P_A sequence	V	T	D	R	S	L
P_B sequence	M	K	A	R	R	D
P_A secondary structures	H	H	H	H	H	H
P_B secondary structures	H	H	H	L	H	H

Figure 4.9. Example of violation of one to one correspondence

The rule for setting E and F matrices is similar, however in this case memory of a previous match must always be kept because there are no pairs stored in these matrices as all pairs stored in B and C matrices are amino acid to gap type. While setting E(i,k), the information of the previous match - whether it's from D(i,k-1), E(i,k-1) or F(i,k-1) – is stored in E(i,k). If E(i,k) can be set from more than one matrix, then the priority is according to the priority scheme in alignment extraction method. This rule is identical for F matrix as well. This rule's reasoning can be observed in Figure 4.7. as well.

The results generated with the new one to one correspondence enforcing scheme are presented in Table 4.13. The similarity function uses the latest version of discrete structure

similarity function, the gap opening penalty is -20 and extension penalty is -2. Also the results obtained by setting the gap opening penalty to -30 and extension penalty set to -3 are presented as well. Again two sets of results are presented to observe the impact of gap penalties on results. However the impact is not very significant on these two particular sets. By comparing these results with best results we have achieved so for, it's interesting to note that 1A17:_ - 1E96.B pair's RMSD has finally been lowered to a satisfactory scale. 1A1Z:_ - 1NTC:A pair's RMSD is lowered in the second set of results. These two pairs' results have been very high from the start. However 1A28:A – 1LBD:_ pair's RMSD has increased because of one to one correspond ace constraint. It can be concluded that forcing one to one correspondence eliminates bad alignments but it can eliminate good paths at the same time.

	Gap opening = -20, extension = -2		Gap opening = -30, extension = -3	
Protein Pair	RMSD (Å)	Length	RMSD (Å)	Length
12AS:A 1PYS:A	7.781368732	210	6.842104912	216
19HC:A 1NEW:_	10.21330643	50	6.905212879	38
1A0A:A 1AM9:A	4.43990612	40	4.43990612	40
1A17:_ 1E96:B	4.164266586	114	3.073358297	114
1A1Z:_ 1NTC:A	12.59808731	55	8.132660866	40
1A28:A 1LBD:_	10.58347034	203	10.24848938	207
1A34:A 1AUY:A	5.899938107	106	4.68641901	120
1A3A:A 1A6J:A	3.048616171	132	3.094862938	133
1A53:A 1NSJ:_	4.346774101	177	4.320028782	179
1A5R:_1 UBI:_	4.349606514	73	4.193257332	75
1A6M:_ 1ASH:_	2.618900776	120	2.618900776	120
1A7T:A 1SML:A	4.57208252	196	5.836570263	203
1A9V:_ 1EHX:A	9.809444427	75	11.85821915	77
1ABA:_ 1ERV:_	5.144833565	69	5.064451218	70
1AC5:_ 1IVY:A	7.325204372	393	6.986800671	393
1ACP:_ 2AF8:_	4.774550915	69	5.149586201	74
1AD3:A 1BPW:A	4.350210667	391	3.580681324	406

Table 4.13. RMSD values of experiment 9

4.11 Experiment 10: Optimization

Over the course of recent experiments, the algorithm has been introduced new parameters (secondary structure similarity) and some parameters' have gotten more complex (gap penalty). Up until this point the general guidelines have been followed in determining parameters, keeping them in similar ranges and changes that have proved their usefulness have been adopted. In our experiments we tried to variate the parameters as much we can manually and for each experiment the best results have been presented out of numerous results sets.

However at this point, manual control and experimentation with parameters is not practical. Besides optimum results may be evading us because we may be overlooking some combinations of parameters. Therefore at this phase, optimization process can be initiated. The optimization will be driven by a genetic algorithm because genetic algorithms are ideal in solving problems by combining solutions' parts to achieve optimum solutions. In our case, the solution sought is a set of parameters, ideal for combination.

The results of optimization are presented in Table 4.14, with the parameter set that yielded the results.

	g_o = -39, g_e = -4 , c_{str} = 0.4, c_{seq} = 0.6, c_{ss} = 1.8		g_o = -18, g_e = -1 , c_{str} = 0.35, c_{seq} 0.65=, c_{ss} = 1.2 & 1-1 correspondence	
Protein Pair	RMSD (Å)	Length	RMSD (Å)	Length
12AS:A 1PYS:A	6.25169	212	6.25169	212
19HC:A 1NEW:_	1.31773	14	1.31773	14
1A0A:A 1AM9:A	4.31527	47	4.31527	47
1A17:_ 1E96:B	9.94836	139	9.94836	139
1A1Z:_ 1NTC:A	12.7107	61	12.7107	61
1A28:A 1LBD:_	6.5578	211	6.5578	211
1A34:A 1AUY:A	4.95136	124	4.95136	124
1A3A:A 1A6J:A	2.7228	136	2.7228	136
1A53:A 1NSJ:_	4.0291	178	4.0291	178
1A5R:_1 UBI:_	2.81389	72	2.81389	72
1A6M:_ 1ASH:_	2.16448	138	2.16448	138
1A7T:A 1SML:A	5.00551	199	5.00551	199
1A9V:_ 1EHX:A	12.6515	61	12.6515	61
1ABA:_ 1ERV:_	5.06445	70	5.06445	70
1AC5:_ 1IVY:A	5.598817	390	7.248859	382
1ACP:_ 2AF8:_	5.14959	74	5.14959	74
1AD3:A 1BPW:A	2.93705	422	2.93705	422

Table 4.14. RMSD values of experiment 10

4.12 Experiment 11: Optimization of Astral40 Dataset

In this experiment, we try to optimize our algorithm based on the results generated by using the Astral40 dataset. On this dataset, we'll use two variations of our algorithm, as explained in methods section. Firstly the same optimization approach of the previous experiment is used; the only difference is the dataset. Eleven proteins were picked from Astral40, belonging to three different families, and each protein is aligned with all members of the family it belongs to.

In this experiment, we also tested our method's capability of domain recognition. The sequences of the protein of this dataset were submitted to ProDom database, which has a utility that discovers domain boundaries of a protein by using a classification scheme dependent on multiple sequence alignments. The idea is to see if the regions of the protein we align are the regions that constitute the domains or not. However the fitness function remains same to see if the domains can be discovered by optimizing for good structural alignments.

	$g_o = -46$, $g_e = -9$, $c_{str} = 0.35$, $c_{seq} = 0.65$, $c_{ss} = 0.4$		CE	
Protein Pair	RMSD (Å)	Length	RMSD (Å)	Length
1NGK:A 1S69:A	3.51889	116	2.2	109
1NGK:A 1IDR:A	2.22581	115	2.2	116
1NGK:A 1DLW:A	2.29497	115	2.3	116
1S69:A - 1IDR:A	1.39818	107	1.2	115
1S69:A - 1DLW:A	1.08984	107	1.1	114
1IDR:A 1DLW:A	0.990477	114	1.0	115
1CSH:_ 1K3P:A	4.55397	315	2.8	344
1CSH:_ 1IOM:A	4.40758	353	2.8	349
1K3P:A 1IOM:A	3.64318	350	2.8	361
1MY6:A 1COJ:A	5.38116	179	2.1	175
1MY6:A 1B06:A	5.3473	193	1.6	176
1MY6:A 1IX9:A	1.76591	194	1.5	194
1COJ:A 1B06:A	2.96567	187	1.7	190
1COJ:A 1IX9:A	5.74693	180	1.6	175
1B06:A 1IX9:A	5.6686	194	1.4	178

Table 4.15. RMSD values of experiment 11

The results are presented in Table 4.15. One to one correspondence constraint is not enforced; the structure similarity function is discrete. Our results are comparable to CE, but for some pairs we fail to find as good alignments as CE does. However all of the alignments have acceptable RMSD values.

With these results, we also checked our ability to discover domains these proteins may share. ProDom was used to discover the domain boundaries of these proteins and they are presented in Table 4.16. The last four proteins contain two domains, and boundaries of each domain is presented, separated '&' symbol. In Table 4.17 we present the boundaries of alignments for each of the two proteins of each alignment. We can observe that boundaries of our alignments correlate with the boundaries discovered by ProDom, in some cases close to %100 correlation. However our method and ProDom do not agree on the same boundaries for 1CSH:_ and 1K3P:A. To decide if this was an error on our part or ProDom's part, we decided to check with another domain detection method, Pfam. Pfam is similar to Prodom, both of the method use sequence alignment method to recognize domains, however Pfam also makes use of profile hidden Markov models. The boundaries discovered by Pfam are also presented in Table 4.16. For 1CSH:_ and 1K3P:A, Pfam reports domain boundaries as 42-420 and 45-408 and these results support our findings. Thus we can see conclude that the error for the 1CSH:_ and 1K3P:A pair comes from ProDom.

Protein	Domain Boundaries of ProDom	Domain Boundaries of Pfam
1NGK:A	2-122	1-121
1S69:A	2-120	1-121
1IDR:A	10-128	13-127
1DLW:A	1-115	1-116
1CSH:_	7-428	42-420
1K3P:A	8-420	45-408
1IOM:A	3-370	4-356
1MY6:A	1-84 & 94-199	1-87 & 91-196
1COJ:A	2-87 & 95-212	1-88 & 90-196
1B06:A	4-90 & 105-210	6-90 & 96-202
1IX9:A	1-86 & 96-205	1-89 & 91-201

Table 4.16. Boundaries from ProDom and Pfam

Aligned Protein Pair	Boundaries of First Protein	Boundaries of Second Protein
1NGK:A 1S69:A	3-121	2-117
1NGK:A 1IDR:A	4-125	14-128
1NGK:A 1DLW:A	4-125	1-115
1S69:A - 1IDR:A	3-113	14-120
1S69:A - 1DLW:A	3-113	1-107
1IDR:A 1DLW:A	14-127	1-114
1CSH:_ 1K3P:A	93-422	86-408
1CSH:_ 1IOM:A	44-425	6-361
1K3P:A 1IOM:A	54-414	15-364
1MY6:A 1COJ:A	18-197	18-198
1MY6:A 1B06:A	1-197	8-205
1MY6:A 1IX9:A	1-198	1-203
1COJ:A 1B06:A	12-201	18-208
1COJ:A 1IX9:A	18-199	17-203
1B06:A 1IX9:A	8-207	1-204

Table 4.17. Boundaries discovered by our Method

4.13 Experiment 12: Optimization of Astral40 Dataset with Algorithmic Variant

In this experiment, we used a variant of our algorithm. This approach is similar to the strip combining approach presented in experiment 8, however the way strips are prepared and combined is different. Similar to the experiment 8, gaps are not permitted. Moreover, mismatches are not allowed, if the similarity of a match is negative, that match is eliminated while the matrix is being prepared. By using these two conditions, we ensure that only very similar regions are aligned.

Since gaps are not allowed, the alignments will be again “strips”. The optimum solution can be discovered by combining non-overlapping strips. In experiment 8, N-best alignments are first obtained from the matrix, then from this set, the alignments are combined with greedy algorithm. This method is slow and wastes computation time because some of the alignments (more than %75 percent) in the set overlap with others and

are discarded. A new and faster method of combining is necessary because during the optimization, a few thousands of alignments are done, making speed is essential.

To solve this problem, we propose an iterative divide and conquer extraction method. The new method of combination is also a greedy algorithm, however the new method doesn't extract unnecessary alignments. Initially, the highest scoring alignment is extracted from the matrix M, after that the matrix M is split into two submatrices, one from the zero to the beginning of extracted alignment, the other submatrix from the end of the extracted alignment to the end of the matrix. This division is presented in Figure 8.

By making the division, we eliminate extraction of alignments that overlap with the pre-extracted alignments, avoiding unnecessary computations. The same procedure is again initiated in the two submatrices: best alignment of the submatrix is found and submatrix is divided into two. If a submatrix is smaller than 5x5 or no alignment can be found in a submatrix, that submatrix is not divided and no search is done in that matrix.

The final solution is the set of the alignments extracted during the divide and conquer method. However before the final solution is prepared, alignments of length lower than five amino acids are discarded to keep the solution pure. Similarities of very short regions may occur at random and possibly can be insignificant. All of the extracted alignments may be combined since divide and conquer method ensures that none of them overlap.

The results generated by using this method are presented in Table 4.17. The discrete structure function is used. This set is the third fittest results, manually picked rather than the first fittest results. Our algorithm is sensitive to small changes in parameters, and the fittest set generates some alignments with RMSD values higher than 5Å. The parameters are also presented in Table 4.17. The combination method picks only very similar regions, thus our alignments are shorter but have lower RMSD values. For some cases, we obtain better superpositions than CE, with only minimal and insignificant changes in length. When the length of the protein increases, it becomes harder to keep vision of global features, the possibility of finding wrong strips increases, thus RMSD values are high.

	c_{str} = 0.2, c_{seq} = 0.8, c_{ss} = 1.0		CE	
Protein Pair	RMSD (Å)	Length	RMSD (Å)	Length
1NGK:A 1S69:A	1.8359	93	2.2	109
1NGK:A 1IDR:A	1.95223	90	2.2	116
1NGK:A 1DLW:A	2.54537	84	2.3	116
1S69:A - 1IDR:A	0.795931	101	1.2	115
1S69:A - 1DLW:A	1.20919	106	1.1	114
1IDR:A 1DLW:A	0.973617	113	1.0	115
1CSH:_ 1K3P:A	3.145	233	2.8	344
1CSH:_ 1IOM:A	5.74582	259	2.8	349
1K3P:A 1IOM:A	2.37883	284	2.8	361
1MY6:A 1COJ:A	4.77745	148	2.1	175
1MY6:A 1B06:A	4.03236	144	1.6	176
1MY6:A 1IX9:A	1.01064	173	1.5	194
1COJ:A 1B06:A	2.4365	159	1.7	190
1COJ:A 1IX9:A	1.67263	134	1.6	175
1B06:A 1IX9:A	4.21922	143	1.4	178

Table 4.18. RMSD values of experiment 12

We also present the boundaries discovered by the combination approach in Table 4.18. The boundaries discovered by this method are similar to boundaries found in previous experiment. The boundaries discovered correlate with Pfam and ProDom except for boundaries of 1CSH:A - 1IOM:A pair. As stated before, combination approach has difficulties when aligning longer proteins because the number of strips increases. The main contribution of combination approach is to lower the RMSD values at the expense of length of the alignment, leading to better superpositions, thus more confident alignments.

Aligned Protein Pair	Boundaries of First Protein	Boundaries of Second Protein
1NGK:A 1S69:A	1-116	1-113
1NGK:A 1IDR:A	2-115	13-119
1NGK:A 1DLW:A	2-121	1-109
1S69:A - 1IDR:A	2-112	13-119
1S69:A - 1DLW:A	2-123	1-115
1IDR:A 1DLW:A	13-127	1-115
1CSH:_ 1K3P:A	61-420	65-408
1CSH:_ 1IOM:A	90-413	44-348
1K3P:A 1IOM:A	53-408	12-356
1MY6:A 1COJ:A	8-197	3-205
1MY6:A 1B06:A	1-184	6-191
1MY6:A 1IX9:A	1-198	1-203
1COJ:A 1B06:A	3-197	3-204
1COJ:A 1IX9:A	17-200	17-205
1B06:A 1IX9:A	6-197	1-197

Table 4.19. Boundaries Defined by Combination Method

4.14 Experiment 13: Domain Recognition between Distant Proteins

In this experiment, the third dataset is used. This database contains pairs of proteins containing ATP-binding, DNA binding and calcium binding domains. Each pair was randomly chosen from its corresponding fold, thus the degree of homology between the proteins are not exact. In this experiment we tested our methods capability of discovering domains among proteins containing the same domain but having different functions.

We ran an optimization procedure for this set of proteins, using combination approach. The discrete structure function was used. The fitness was again targeted at the combination of RMSD and length. However since combination approach picked very similar regions between the proteins, our algorithm converged to parameter sets that generated very short solutions with very small RMSD values. Out of the result set, we picked a parameter set that yielded longer solutions. The results are presented in Table 4.19

Aligned Protein Pair	Boundaries of First Protein	Boundaries of Second Protein
1K04:A - 1JOY:A	9-132	2-60
1ZBD:B - 1WEM:A	14-118	46-72
1QBJ:A - 2HDC:A	2-64	7-78
1UWO:A - 5CPV:_	26-77	5-106
1R0O:A -1RMD:_	2-64	25-55

Table 4.20 Boundaries by our method results

In Table 4.20. the domain predictions of these proteins from Pfam are presented. Most of the boundaries of our alignments fall in the range specified by Pfam, however for some pairs we may align regions outside the boundaries. For example, 1K04:A of the first pair is aligned more than the Pfam boundary. The same case holds for 1ZBD:B whereas the alignment of 1WEM:A is longer. However we roughly align the similar regions. The error may occur because these proteins are distant relatives (they come from different super families of SCOP).

Protein	Domain Start	Domain End
1K04:A	24	162
1JOY:A	11	66
1ZBD:B	2	134
1WEM:A	18	70
1QBJ:A	7	73
2HDC:A	2	97
1UWO:A	3	80
5CPV:_	43	109
1R0O:A	9	80
1RMD:_	26	64

Table 4.21 Boundaries by Pfam

5 CONCLUSION

In this section, a summary of the development process of our algorithm and the results is presented. Furthermore, the list of the software applications that were developed to implement our algorithm is provided. Lastly, the future direction of our research and other potential uses of our algorithm are explained.

5.1 Summary

New and fast algorithms to process and derive information from protein structure data are needed with the increasing number of protein structures deposited in the PDB. Protein structure information is especially useful to compare remote homologs because sequence alignment methods fail to discover similarities among these kinds of proteins. We designed an alignment algorithm that uses both sequence and structure data to make it capable of finding accurate alignments even for proteins of minimal sequence identity.

We designed our algorithm based on a well established alignment approach, dynamic programming. Dynamic programming approach ensures a quick search of the global optimum. The similarity function that drives the alignment algorithm combines primary, secondary and tertiary structure similarities. Affine gap penalty scheme was preferred to achieve more accurate alignments.

The measure we used to evaluate the accuracy of our alignments is RMSD measure. We used different combinations of the three similarity parameters and tried different functions to evaluate the tertiary structure similarity, while we used similarity scoring matrices for primary and secondary structure similarities. The different combinations and

functions were tried to achieve longest alignments with lowest RMSD scores. In building our tertiary structure similarity score, we observed the alignments built by other structure alignment methods and tried to set the scoring criteria of our function accordingly.

Variations of the general dynamic programming approach were also tried. We tried building the optimum alignment by combining short gapless alignments and short highly similar alignments. We also introduced different constraints to dynamic programming approach, forcing one to one correspondence between secondary structure elements. We also tried to optimize all the different parameters we introduced by using a genetic algorithm driven optimization process.

Our algorithm discovers regions that share high sequence and structure similarity. We expected these regions to belong the domains that proteins may have when we are aligning two proteins that both have the same domain. We explored this possibility as well, and checked if the regions we align agree with other domain boundary recognition methods.

5.2 Discussion

In this section, we discuss the results of our algorithm in terms of speed, accuracy and novelty. We'll compare our method to existing ones and explain the advantages and shortcomings.

In terms of speed, our method is comparable to methods designed for speed, like FAST. FAST reports that a typical alignment takes approximately 1 second with a PIII 1.2GHz, whereas more computational intensive methods like CE take a few seconds. Our algorithm also takes less than a second when performing a typical alignment with a PIII 1.7GHz, even though the code hasn't been optimized for speed. By using premade contact maps and network models, our method can be used for fast database searches.

The factor that makes our algorithm faster than most methods is the model we use. By representing the large set of three dimensional coordinates with a graph, and using two

structural comparison parameters obtained from the graph with quick and straightforward calculations, the structural similarity can assessed very quickly compared to other methods that make complex distance calculations during their alignment process. The similarity matrices we use for primary and secondary structures also help to increase the speed of our algorithm, because using them reduces the process of assessing the primary and secondary structure similarity to a matrix look-up operation.

The accuracy of our alignments is comparable to other methods for most cases. We used CE for structural alignment comparison process since it's a widely accepted and still one of the best alignment methods available. The RMSD and the length of an alignment are two parameters that can assess the goodness of a structural alignment; RMSD should be as small as possible while maximum number of amino acids should be aligned. Generally we perform comparably, however for most cases CE is better. This can be attributed to the trade off between speed and accuracy, by modeling the three dimensional structure, some details of the model may be overlooked while reducing computational complexity. Moreover, our method doesn't calculate inter atomic distances of aligned amino acids during the alignment, and doesn't do post processing on the alignment to further reduce RMSD unlike CE. By sacrificing speed and increasing computational complexity, such processes can also be introduced to our method, reducing the RMSD values of our results.

Using two different approaches to obtain the final alignment can also cause another tradeoff situation for our method. The first and main approach is more tolerant of mismatches and uses gaps to maximize the length of the alignment, causing some bad matches to occur. The second approach, combination approach doesn't allow mismatches or gaps, thus the alignments found by this approach contain no or little bad matches. Given these, the alignments generated by the second approach are more accurate, with lower RMSD values compared to the alignments of the first method, but they are shorter. The second approach can be used for aligning highly similar proteins to pinpoint highly similar regions, whereas the first approach has a more general use and can be applied for proteins with variable degrees of similarity.

It was observed that our method failed to generate alignments with acceptable RMSD values for some pairs for protein pairs. To discover the problem, we checked the alignments of these pairs generated by CE and calculated the similarity scores of matched amino acids using our similarity scoring function. It was observed that our similarity scoring function cannot assess the similarity accurately for these pairs. We noted that the structure of pairs we fail to align accurately were determined by using NMR. NMR method is less accurate compared to X-ray crystallography and since we are using just a single atom (C_β or C_α) and a cutoff distance to evaluate contacts, the accuracy of PBD files is critical. Even an error of lower than 1Å can result in loss of some contacts, resulting in a wrong contact map. Thus we suspect that the quality of the PDB files may be affecting our results.

The accuracy of the domain discovery approach is satisfactory. The regions aligned by our algorithm correlate with the regions predicted as domains by ProDom and Pfam. In some cases, results of ProDom and our method differ, but Pfam correlates with our results. Unlike Prodom, Pfam also uses structural information to discover domains, thus we can say that our method performs better than Prodom and is on par with Pfam.

The novelty of our approach is the use of the network (graph) representation of protein in an alignment algorithm. The network model has been applied and studied before by Atilgan et.al. However we have used this model to capture the structural properties of proteins and use this model for comparison purposes, allowing us to align similar regions of two proteins. Since the network model is based on graph theory, we are actually using graphs in an alignment method and modeling the alignment as a graph matching problem.

The network model contains the primary, secondary and tertiary structural properties of the protein. Each node, representing one amino acid, has different attributes: the type of the amino acid representing primary information, the type of the secondary structure element representing secondary structure and the cliquishness and connectivity of that node, representing the tertiary structure. By using these three parameters together, we have achieved better results than using them separately, and other different secondary structure

alignment methods. Using each parameter singularly for aligning the proteins of the first dataset yields a best of 6.0Å average RMSD by using secondary structure. Primary structure information also yields 6.0Å average but find shorter alignments whereas tertiary structure yields longer alignments but an average RMSD of 7.0 Å. Alignments obtained by using secondary and primary structure information have an approximate average of 5.0Å and the average length is almost half of alignments discovered by other alignment methods like CE. Our method, which uses all three parameters, achieves 4.2 Å RMSD average while the average alignment length is very slightly larger than average attained by CE.

The constraint enforcement procedure we introduced into our dynamic alignment algorithm is also another novel addition to alignment routine. Methods aligning secondary structures generally force one to one correspondence between secondary structures by post processing the alignment. We have integrated additional matrices to keep track of the matches being made and eliminated matches that violated one to one correspondence.

5.3 Applications Developed

In this section, the software applications developed during the research are presented.

PDB Parser: This application is used to parse PDB files and store information inside the files. PBD files contain lots of information with a specific format, thus the parser can be used as a general purpose tool to extract and display desired content. The parser processes primary, secondary and tertiary structure information and stores them.

Protein Class: A software package was prepared to implement the amino acid network model. The package is capable of processing the data obtained by the PDB parser and prepare contact maps. After contact map is prepared, the model can also be prepared and the connectivity and cliquishness parameters can be obtained.

Alignment Tools: An extensive alignment algorithm was coded, capable of different kinds of alignments. The purpose of the alignment tool is to implement our methods, however sequence alignment and secondary structure alignments can be done separately as well. The algorithm can also use affine gaps and do global and local alignments. We have also integrated different methods of extracting alignments, as explained in the results section. The one to one correspondence constraint is also implemented in this package.

Genetic Optimizer: A general purpose alignment parameter optimizer has been prepared to optimize our results. It can also be used to optimize other alignments, if a proper fitness function is defined.

RMSD Calculation Scripts: A group of scripts to automate the RMSD calculation of an alignment were prepared using Tcl-Tk language. The scripts make use of the VMD software to perform the superposition and RMSD calculation.

5.4 Future Directions

The results of our algorithm have shown promise in the two purposes it was designed for. Structural alignment results have obtained a good balance between speed and accuracy, we can discover acceptably accurate alignments which are comparable to existing methods and our algorithm performs faster. The accuracy can be further improved by using more than one contact map, or making the contact map define contacts continuously, rather than a binary true/false.

The alignment algorithm also scores every alignment it finds, thus can calculate the distance between each protein. The higher the alignment score is, the closer two proteins are. The scores of the alignments can be used for fold classification purposes, we can predict the members of same fold (family) by comparing the scores of alignment.

We have tried fold classification on the dataset of Capriotti et.al, by aligning each protein with all others and grouped highest scoring protein pairs together as homologs after

a score normalization process. The results are promising with more than %80 percent accuracy. Tests on larger datasets can be run to explore the full capability of this possible use of our algorithm.

Our alignment combination approach has some problems when working with long proteins (around 400 amino acids). The difficulty arises from the fact that there are too many local similarities between two such long proteins, especially if its motifs are repetitive, thus we may lose sight of global order when working with local similarities. To address this problem, the address this problem, both of the methods we devised can be used together. By using our first alignment method, by allowing gaps mismatches, we can extract one alignment which captures the global picture. Then based on this first alignment, we can extract short strips of high similarity, without gaps or mismatches, and combine them according to the first alignment. This method shouldn't increase run times dramatically, and can achieve more confident results than each of the separate methods.

REFERENCES

[1]. Branden, C. and Tooze, J. (1991) Introduction to Protein Structure. New York: Garland Publishing, 62-63.

[2]. Nelson, D.L., Cox, M.M. (2005) Lehninger Principles of Biochemistry, Fourth Edition. New York: W. H. Freeman & Co, 43-44.

[3]. Gromiha, M.M., Saraboji, K., Ahmad, S., Ponnuswamy, M.N. and Suwa, M. (2004) Role of non-covalent interactions for determining the folding rate of two-state proteins. Biophys. Chem. 107, 263-72.

[4]. Abkevich, V.I. and Shakhnovich, E.I. (2000) What can disulfide bonds tell us about protein energetics, function and folding: Simulations and bioinformatics analysis. J. Mol. Biol. 300, 975-985.

[5]. Kabsch, W., Sander, C. (1983) Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 22, 2577-2637.

[6]. Ponting, C., Russell, R.R. (2002) The Natural History of Protein Domains. Annual Review of Biophysics and Biomolecular Structure. 31, 45-71.

[7]. Holm, L., Sander, C. (1998) Dictionary of recurrent domains in protein structures. Proteins: Structure, Function, and Genetics. 33, 88-96.

[8]. Kendrew, J.C., Bodo, G. Dintzis, H.M., Parrish, R.G., Wykcooff, H., Phillips, D.C. (1958) A three-dimensional model of the myoglobin molecule obtained by X-rat analysis. Nature.

[9]. Martin, G.,E; Zekter, A.,S. (1988) ‘’Two-Dimensional NMR Methods for Establishing Molecular Connectivity’’; VCH Publishers, Inc: New York. 59

[10]. Drenth J. (1999) *Principles of Protein X-Ray Crystallography.* Springer-Verlag Inc. New York.

[11]. Wuthrich, K. (1986) NMR of Proteins and Nucleic Acids. John Wiley & Sons, New York. 166.

[12]. Moult, J., Fidelis, K., Zemla, A., Hubbard, T. (2003) Critical assessment of methods of protein structure prediction (CASP)-round V. Proteins: Structure, Function, and Genetics. 53, 334-339.

[13]. http://www.wwpdb.org/documentation/format2.3-0108-us.pdf (retrieved on Jan. 28, 07)

[14]. Bernstein, F.C., Kowtzle, T.F., Williams, G.J., Meyer, E.F., Brice, M.D. Rodgers, J.R., Shimanouchi, T., Tasumi, M. (1977) The Protein Data Bank: A computer-based archibal file for macromolecular structures. Eur. J. Biochem. 80, 319-324.

[15] Altschul S.F., Gish W., Miller W. Myers E.W., Lipman D.J (1990) Basic Local Alignment Search Tool. *J. Mol . Bio*. **215**, 403-410.

[16] Thompson J.D., Higgins D.G., Gibson T.J. (1994) CLUSTAL W: Improving the Sensitivity of Progressive Multiple Sequence Alignment Through Sequence Weighting, Positions-Specific Gap Penalties and Weight Matrix Choice. *Nucleic Acids Res*, **22**, 4673-4680.

[17] Schwarz R., Dayhoff M. (1979) Matrices for Detecting Distant Relationships. *Atlas of Protein Sequences*, **5**.

[18] Henikoff S., Henikoff J.G. (1992) Amino Acid Substitution Matrices from Protein Blocks. *PNAS*, **89**, 10915-10919.

[19] Holm L., Sander C. (1996) The FSSP Database: Fold Classification Based on Structure-Structure Alignment of Proteins. *Nucleic Acids Research*, **24**, 206-209.

[20] Smith T.F., M.S. Waterman. (1981) Identification of Common Molecular Subsequences. *J. Mol. Biol.* **147**, 195-197.

[21] Needleman S., Wunsch C. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*. **48(3)**, 443-53.

[22] Lipman D.J., Pearson W.R. (1985) Rapid and Sensitive Protein Similarity Searches. *Science*, **227**, 1435-1441

[23] Chothia C., Lesk A.M. (1986) The Relation Between the Divergence of Sequence and Structure in Proteins. *EMBO J.*, **5**, 823–826.

[24] Taylor W.R. (1999) Protein Structure Comparison Using Iterated Double Dynamic Programming. *Protein Sci.*, **8**, 654-665.

[25] Kabsch W. (1978) A Discussion of the Solution for the Best Rotation to Relate Two Sets of Vectors. *Acta Crystallogr. A*, **34**, 827–828.

[26]Eidhammer I., Jonassen I., Taylor W.R. (2000) Structure Comparison and

Structure Patterns. *J Comp. Bio.,* **7**, 685–716.

[27] Zhu J., Weng Z. (2005) FAST: A Novel Protein Structure Alignment Algorithm. *Proteins: Structure, Function and Bioinformatics*, **58**, 618-627.

[28] Jewett A.I., Huang C.C., Ferrin T.E. (2003) MinRMS: An Efficent Algorithm for Determining Protein Structure Similarity Using Root-Mean-Squared-Distance. *Bioinformatics*, **19**, 625-634.

[29] Shindyalov I. N., Bourne P.E. (1998) Protein Structure Alignment by Incremental Combinatorial Extension (CE) of the Optimal Path. *Protein Engineering*, **11**, 739-747.

[30] Holm L., Sander C. (1993) Protein Structure Comparison by Alignment of Distance Matrices. *J Mol Bio*, **233**, 123-138.

[31] Murzin A.G., Brenner S.E., Hubbard T., ChothiaC. (1995) SCOP:A Structural Classification of Proteins Database for the Investigation of Sequences and Structures. *J. Mol. Biol.*, **247**, 536–540.

[32] Orengo C.A., Michie A.D., Jones S., Jones D.T., Swindells M.B. Thornton J.M. (1997) CATH—A Hierarchic Classification of Protein Domain Structures. *Structure*, **5**, 1093–1108.

[33] Moult J., Fidelis K., Zemla A., Hubbard T. (2003) Critical Assessment of Methods of Protein Structure Prediction (CASP)-round V. *Proteins*, **53**, 334–339.

[34] Skolnick J., Fetrow J.S., Kolinski A. (2000) Structural Genomics and Its Importance For Gene Function Analysis. *Nat. Biotechnol.*, **18**, 283–287.

[35] Baker D., Sali A. (2001) Protein Structure Prediction and Structural Genomics. *Science*, **294**, 93–96.

[36] Corpet F., Gouzy J, Kahn D. (1998) The ProDom Database of Protein Domain Families. Nucleic Acids Research, **26**, 323-326.

[37] Bairoch A. Apweiler R. (1997) The SWISS-PROT Protein Sequence Database: Its Relevance to Human Molecular Medical Research. *J. Mol. Med.*, **75**, 312-316.

[38] Bateman A., Birney E., Cerruti L., Durbin R., Etwiller L., Eddy S.R., Griffiths-Jones S., Hoew K.L., Marshall M., Sonnhammer E.L.L. (2002) The Pfam Protein Families Database. *Nucleic Acids Research*, **30**, 276-280.

[39] Atilgan A.R., Akan P., Baysal C. (2004) Small World Communication of Residues and Significance for Protein Dynamics. *Biophysical Journal*, **86**, 85-91.

[40] Vendruscolo M., Kussell E., Domany E. (1997) Recovery of Protein Structure from Contact Map. *Folding&Design*, **2(5)**, 295-306.

[41] Wallqvist A., Fukunishi Y., Murphy L.R., Fadel A. Levy R.M. (2000) Iterative Sequence/Secondary Structure Search for Protein Homologs: Comparison with Amino Acid Sequence Alignments and Application to Fold Recognition in Genome Databases. *Bioinformatics*, **16**, 988-1002.

[42] Pascarella S. Milpetz F. Argos P.A. (1996) Databank (3D_ali) Collecting Related Proteins Sequences and Structures. Protein Engineering, 9, 249-251

[43] Marchler-Bauer A., Anderson J.B., P.F. Cherukuri P.F., DeWeese-Scott C., Geer L.Y., Gwadz M., He S., Hurwitz D.I. , Jackson J.D., Ke Z., Lanczycki C.J., Liebert C.A., Liu C., Lu F., Marchler G.H., Mullokandov M., Shoemaker B.A., Simonyan V., Song J.S., Thiessen P.A., Yamashita R.A., Yin J.J., Zhang D., Bryant S.H.(2005) CDD: a Conserved Domain Database for Protein Classification. *Nucleic Acids Research*, **33**, 192-196.

[44] Altschul S.F. (1989) Gap Costs for Multiple Sequence Alignment. *J Theor Biol*, **138**, 297-309.

[45] Capriotti E., Fariselli P., Rossi I., Casadio R. (2003) A Shannon Entropy Based Filter Detects High-Quality Profile-Profile Alignments in Searches for Remote Homologs. *Proteins: Structure, Function and Bioinformatics*, **54**, 351-360.

[46] Brenner S.E., Koehl P., Levitt M. (2000) The ASTRAL Compendium for Protein Structure and Sequence Analysis. *Nucleic Acids Research*, **28**, 254-256.

[47] http://astral.berkeley.edu/