Amino Acid Interaction Network Prediction in Protein using Multi-objective Evolutionary Algorithm

Exam Roll: 1718

submitted in partial fulfilment of the degree of

Master of Science

at the University of Dhaka, Dhaka,

Bangladesh.

10 October 2013

Abstract

Protein can be represented by amino acid interaction network. This is a graph whose vertices are the proteins amino acids and whose edges are the interactions between them. This interaction network is the first step of proteins three-dimensional structure prediction. The network can be predicted using multi-objective evolutionary algorithm and the interaction between amino acid can be confirmed using ant colony algorithm optimization which is a probabilistic optimization algorithm. In this thesis a new multi-objective evolutionary optimization algorithm has been proposed to predict protein secondary structure network using ant colony optimization approach to predict the amino acid interactions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Proteins are biological macromolecules performing a vast array of cellular functions within living organisms. The roles played by proteins are complex and varied from cell to cell and protein to protein. The best known role of proteins in a cell is performed as enzymes, which catalyze chemical reaction and increase speed several orders of magnitude, with a remarkable specificity. And speed of multiple chemical reactions is essential to the organism survival procedures like DNA replication, DNA repair and transcription. Proteins are storage house of a cell and transports small molecules or ions, control the passages of molecules through the cell membranes, and so forth. Hormone, another kind of protein, transmits information and allow the regulation of complex cellular processes.

Genome sequencing projects generate an ever increasing number of protein sequences. For example, the Human Genome Project has identified over 30,000 genes [1] which may encode about 100,000 proteins. One of the first tasks when annotating a new genome is to assign functions to the proteins produced by the genes. To fully understand the biological functions of proteins, the knowledge of their structure is essential.

Proteins are amino acids chain bonded together in peptide bonds, and naturally adopt a native compact three-dimensional form. The process of forming three-dimensional structure of a protein is called protein folding and this is not fully understood yet in System Biology. The process is a result of interaction between amino acids which form chemical bond to make protein structure.

In this research paper, we are going to see how protein structures are related with interaction network of amino acid and we will propose a new algorithm to predict a interaction network of amino acids using two new emerging optimization techniques,

multi-objective optimization based on evolutionary clustering and ant colony optimization.

1.1 Objective

Uses of machine learning approaches like artificial neural network, support vector machine (SVM) and Baysian network in predicting protein structure was successful but in cost. These processes are too much complex and time consuming. Time complexities of almost all of these approaches are exponential. This lacking draws the attention of predicting protein structure in evolutionary field. Because complexities of evolutionary approaches like genetic algorithm can be controlled through the number of intended generations.

In [2], Gaci *et al.* used genetic algorithm to predict the incident network of amino acids but their fitness function is just the distance between the amino acid atoms. But only distance could not define the interaction between amino acid. There are several other attribute like similarity in amino acid structure, hydrophilicity or hydrophobicity and different torsion angles which are described in chapter 2 can affect the interaction. Gaci *et al.* [2] also used ant colony approach to predict the interaction of amino acid. But it is a time consuming approach and have more chance to stuck in local minima or maxima.

We can represent a protein as a network of amino acid and these amino acid interact with each other and forms a three dimensional structure. Objective of this research is find and efficient algorithm to predict the amino acid interaction network using a multi-objective evolutionary algorithm [3], [4] and a probability based ant colony optimization approach. The multi-objective evolutionary approach is to consider all the attributes which affect interaction in amino acid network including distance.

1.2 Motivation

We can treat proteins as networks of interacting amino acids [5], a directed graph, where amino acids are vertices and interaction between them are edges of the graph.

The three–dimensional structure of a protein is represented by the coordinates of its amino acid atoms. The protein Data Bank (PDB) [6] contains all these information and regroups all experimentally solved protein structures. We can easily compute the distance between two amino acids and different torsion angles between atoms for example,

ϕ and ψ angles which define affinity. This distance and affinity predicts how these amino acids interact with each other. If there are N amino acid to consider, we can make a 0 1 matrix of size $N \times N$, which is called incident matrix. The incident matrix represents a graph of N vertices. From this incident matrix, the three–dimensional structure of the protein can be predicted. There are some other attributes like hydrophobicity and hydrophilicity of amino acid also affect the interaction of amino acids.

1.3 Overview of this book

In this chapter we have introduced the problem of predicting amino acid interaction network in protein. Rest of the chapters are organized as follows.

In Chapter 2, we will acquire some background knowledge by discussing about protein structure including amino acid, primary, secondary and tertiary structure of protein.

In Chapter 3, we are going to discuss about some existing researches on protein structure prediction and amino acid interaction network prediction.

In Chapter 4, we will discuss about amino acid interaction network and verify network properties of amino acid in protein with PDB data.

In Chapter 5, we are going to formulate the amino acid interaction network prediction problem theoretically and mathematically.

In Chapter 6, we will present a new algorithm based on multi-objective optimization and ant colony optimization to predict the formulated problem of amino acid interaction network prediction.

In Chapter 7, we will analyze the algorithm with some PDB data and show the result.

Finally, Chapter 8 concludes the document.

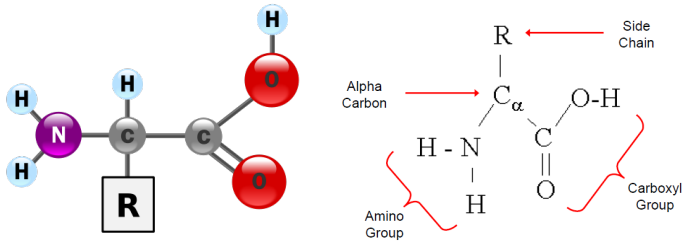
Chapter 2

Background Study

Base of a good research is the understanding of the background terms and definition. To understand the amino acid interaction network and its function in protein structure, one have to clearly understand about protein structure. In this chapter as background knowledge discovery we will discuss about amino acid and protein and its different types of structures.

2.1 Amino Acid

Amino acids are the building blocks of proteins. Protein is nothing but sequences of amino acids linked by peptide bonds. Amino acids are one of the most biologically important organic compounds made from amine ($-NH_2$) and carboxylic acid ($-COOH$) functional groups, along with a side–chain specific to each amino acid. The properties



(a) Amino Acid (b) Amino acid with different group

Figure 2.1: Basic Structure of Amino Acid

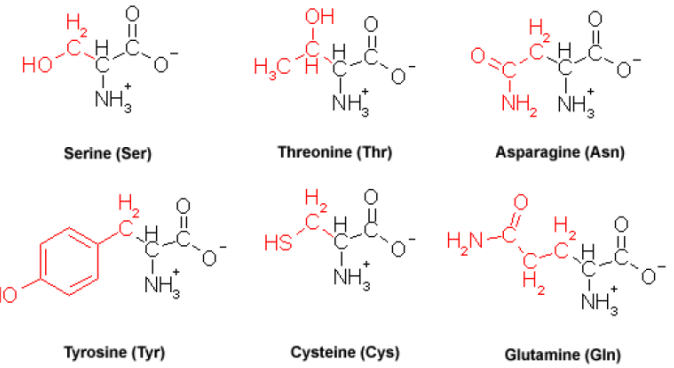
of each amino acid are determined by its specific side chain. After water, amino acids comprise the second largest component of human muscle, cells and other tissues in the

form of proteins. The key components of an amino acid are carbon, hydrogen, oxygen and nitrogen, other elements are found in the side–chains of certain amino acids. Though, according to [7], about 500 amino acids are known, there are 20 different, naturally occurring amino acids. Amino acid names are often abbreviated as either three letters or single letter.

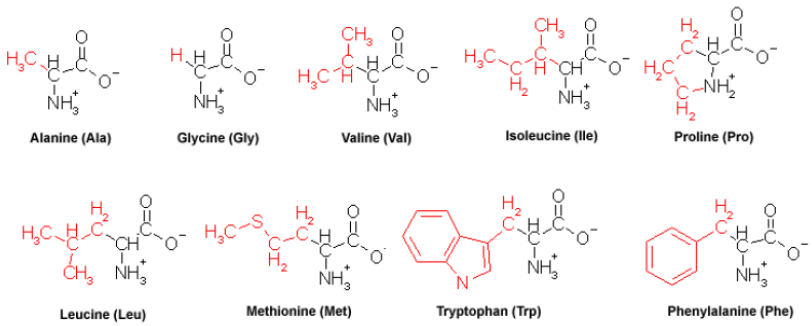
- Alanine (Ala / A)
- Arginine (Arg / R)
- Asparagine (Asn / N)
- Aspartic Acid (Asp / D)
- Cysteine (Cys / C)
- Glutamic Acid (Glu / E)
- Glutamine (Gln / Q)
- Glycine (Gly / G)
- Histidine (His / H)
- Isoleucine (Ile / I)
- Leucine (Leu / L)
- Lysine (Lys / K)
- Methionine (Met / M)
- Phenylalanine (Phe / F)
- Proline (Pro / P)
- Serine (Ser / S)
- Threonine (Thr / T)
- Tryptophan (Trp / W)
- Tryosine (Tyr / Y)
- Valine (Val / V)

Each amino acid has the same fundamental structure, differing only in the side–chain, designated by R–group. The carbon atom to which the amino group, carboxyl group, and side chain (R–group) are attached is the alpha carbon (C_α). The alpha carbon is the common reference point for coordinates of an amino acid structure. The side chains vary in shape, size, charge and polarity. Due to the variety in side chains, properties of amino acid vary. Among the 20 amino acids, Serine (Ser), Threonine (Thr), Asparagine (Asn), Tyrosine (Tyr), Cysteine (Cys) and Glutamine (Gln) are polar residue and Alanine (Ala), Glycine (Gly), Valine (Val), Isoleucine (Ile), Proline (Pro), Leucine (Leu), Methionine (Met), Tryptophan (Trp) and Phenylalanine (Phe) are non–polar residue. The side chains of polar amino acids have partial positive and negative charges and are attracted to water and found mostly in the surface of protein. On the other hand, there are three amino acids that have basic side chains at neutral pH. These are arginine (Arg), lysine (Lys), and histidine (His). Their side chains contain nitrogen and resemble ammonia, which is a base. Their pKa's are high enough that they tend to bind protons, gaining a positive charge in the process. Two amino acids have acidic side chains at neutral pH. These are aspartic acid or aspartate (Asp) and glutamic acid or glutamate (Glu). Their side chains have carboxylic acid groups whose pKa's are low enough to lose protons, becoming negatively charged in the process. The nine non–polar amino acids are hydrophobic. Side chains of these amino acids are composed mostly of carbon and hydrogen, have small dipole moments, and tend to be repelled from water. This fact has important implications for proteins tertiary structure. Most protein molecules have a hydrophobic core, which is not accessible to solvent and a polar surface in contact with the environment. While the core is built up with hydrophobic amino acid residues, polar and charged amino acids preferentially cover the surface of the molecule and are in contact with solvent due to their ability to form hydrogen bonds. Very often they also interact with each other : positively and negatively charged amino acids form so called salt bridges, while polar amino acid side chains may form side chain–side chain or side chains–main chain hydrogen bonds (with polar amide carbonyl groups). It has been observed that all polar groups capable of forming hydrogen bonds in proteins do form such bonds. Since these interactions are often crucial for the stabilization of the protein three–dimensional structure, they are normally conserved. In the figure 2.3 we can see some statistics on the distribution of the different amino acids within protein molecules.

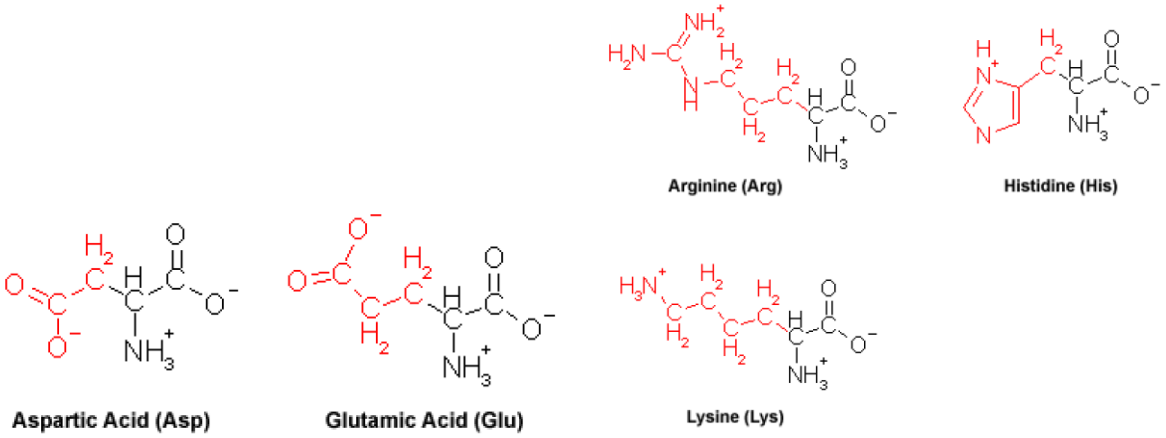
The figure 2.4 demonstrates that while a high fraction of the hydrophobic amino acids are buried within the core of the structure, this number gradually decreases for



(a) Polar Residue



(b) Non–Polar Residue



(c) Acidic Residue

(d) Basic Residue

Figure 2.2: Amino Acid residue with different property

amino acids with polar groups and reaches a minimum for charged residues (the vertical axis shows the fraction of highly buried residues, while the horizontal axis shows the amino acid names in one–letter code).

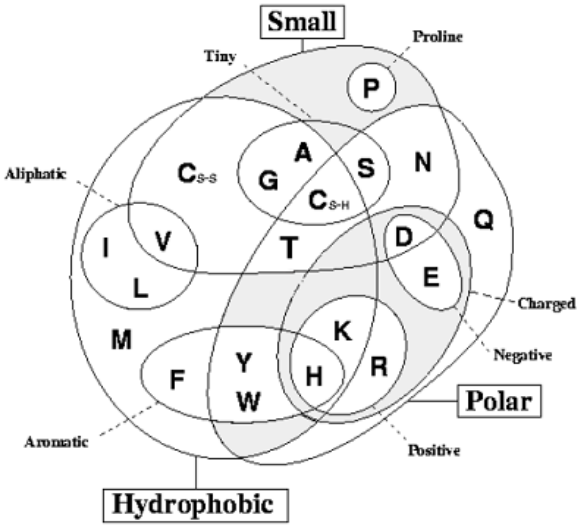


Figure 2.3: Amino Acid Property Diagram

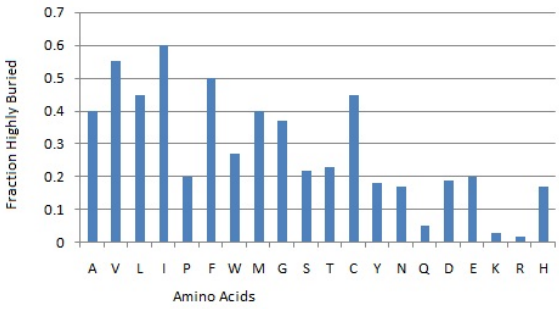


Figure 2.4: Different types of amino acid with fraction buried in protein

2.2 Protein Structure

To make a protein, these amino acids are joined together in a polypeptide chain through the formation of a peptide bond. A peptide bond or amide bond is a covalent chemical bond formed between two molecules when the carboxyl group of one molecule reacts with the amino group of the other molecule, causing the release of a molecule of water (H_2O), hence the process is a dehydration synthesis reaction (also known as a condensation reaction), and usually occurs between amino acids. Inside cells, peptide bonds are formed within ribosomes (a macromolecule inside cell) during a process called translation. During protein synthesis or translation, amino acids are covalently bonded to each other through a peptide bond as in the figure 2.5. Peptide chains that are less than 40–50 amino acids or residues are often referred to as a polypeptide chains since they are too small to form a functional domain. Larger than this size, they are

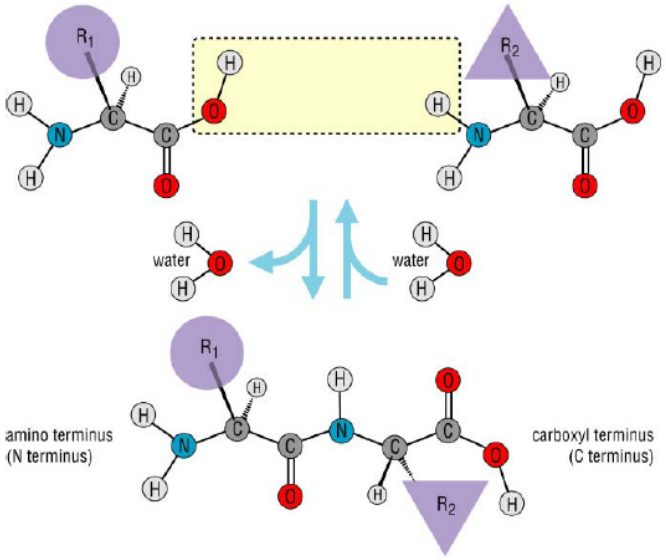


Figure 2.5: Formation of peptide bond

called proteins. Proteins are nothing more than long polypeptide chains.

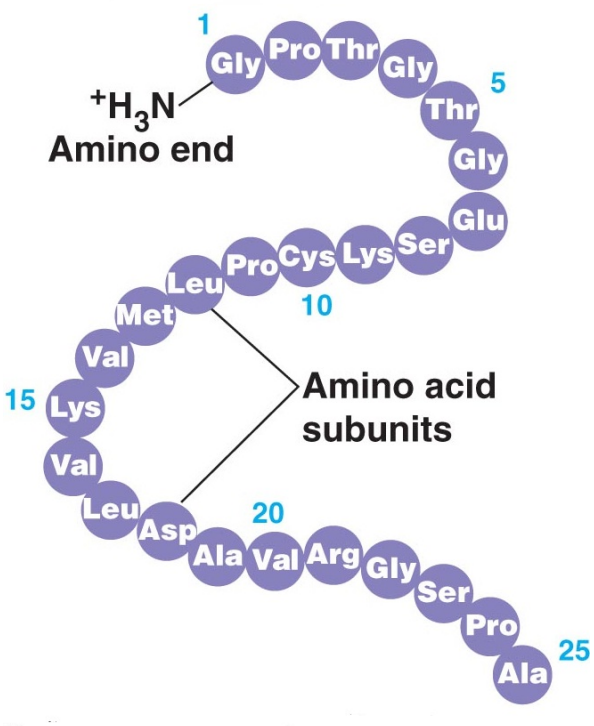
2.2.1 Protein Primary Structure

The structure, function and general properties of a protein are all determined by the sequence of amino acids that makes up the primary sequence. The primary structure of a protein is the linear sequence of its amino acid structural units and it is a part of whole protein structure. Though polypeptides are unbranched polymers, so their primary structure can often be specified by the sequence of amino acids along their backbone. But proteins can become cross–linked, most commonly by disulfide bonds.

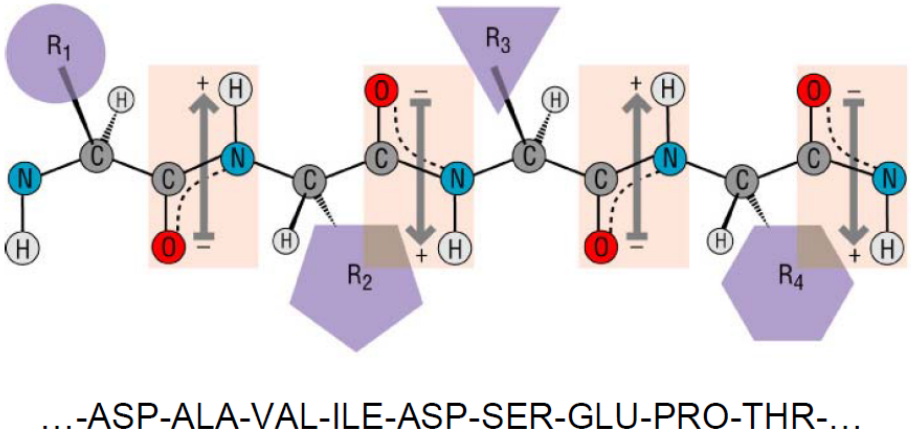
2.2.2 Protein Secondary Structure

Torsion Angle

The two torsion angles of the polypeptide chain, also called Ramachandran angles, describe the rotations of the polypeptide backbone around the bonds between $N - C\alpha$ (called Phi, ϕ) and $C\alpha - C$ (called Psi, ψ). Torsion angles are among the most important local structural parameters that control protein folding –essentially, if we have a way to predict torsion angles for a particular protein, we would be able to predict its 3D structure. The reason is that these angles provide the flexibility required to for folding the polypeptide backbone, since the third possible torsion angle within the protein backbone (called omega, ω) is essentially flat and fixed to 180 degrees. This is



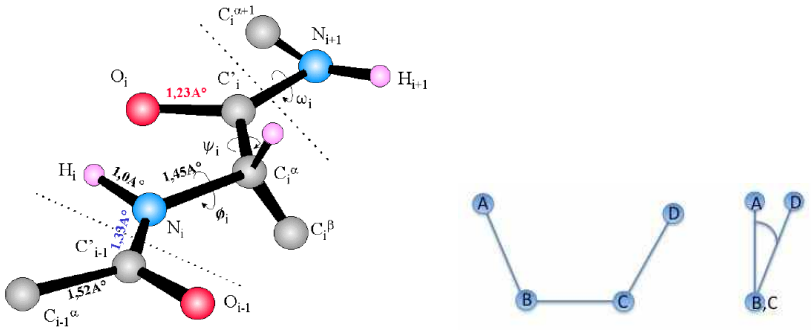
(a) Amino acid chain as protein primary structure



…-ASP-ALA-VAL-ILE-ASP-SER-GLU-PRO-THR-…

(b) Amino acid's are connected with peptide bond in protein primary structure

Figure 2.6: Protein Primary Structure



(a) Torsion Angle (b) Dihedral Angle

Figure 2.7: Torsion Angles

due to the partial double–bond character of the peptide bond, which restricts rotation around $C-N$ bond, placing two successive alpha–carbons and C, O, N and H between them in one plane. Thus rotation of the main chain or backbone of a protein can be described as the rotation of the peptide bond planes relative to each other. Torsion angles are dihedral angles, which defined by 4 points in space. In proteins the two torsion angles ϕ and ψ describe the rotation of the polypeptide chain around the two bonds on both sides of the alpha carbon atom. The standard IUPAC definition of a dihedral angle is illustrated in the figure 2.7b. A, B, C and D illustrated the position of the 4 atoms used to define the dihedral angle. The rotation takes place around the central $B-C$ bond. The view on the right is along the $B-C$ bond with atom A placed at 12 oclock. The deviation of the $A-B$ and $C-D$ bonds from each other is measured by the deviation of D from A, where positive angle referred to as a clockwise rotation. The Ramachandran plot provides an easy way to view the distribution of torsion angles of a protein structure. It also provides an overview of allowed and disallowed regions of torsion angle values, serving as an important factor in the assessment of the quality of protein three–dimensional structures. The torsion angels in proteins are restricted to certain values, since some angles will result in sterical clashes between main chain and side chain atoms in polypeptide. For each type of the secondary structure elements there is a characteristic range of torsion angle values, which can clearly be seen on the Ramachnadran plot: on the left plot the region marked α is for alpha–helices and β is for beta–sheet. The horizontal axis on the plot are ϕ value, while the vertical shows ψ values. Each dot on the Ramachandran plot shows the ϕ and ψ values for an amino acid in a protein. Notice that the counting in the left hand corner starts from -180 and extends to $+180$ for both vertical and horizontal axes. This is a convenient

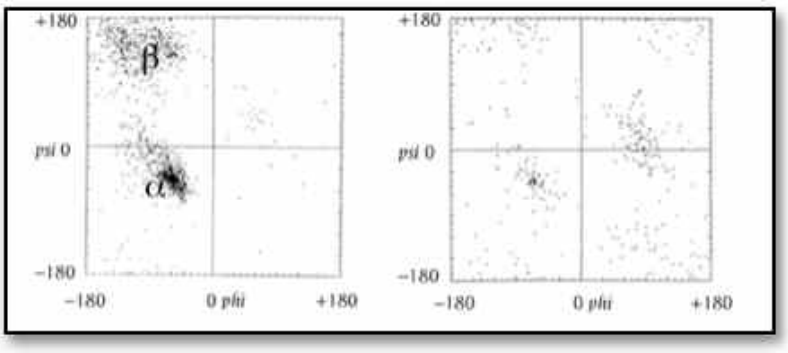


Figure 2.8: Ramachandra Plot

presentation and allows clear distinction of the characteristic regions of $\alpha-$helices and $\beta-$sheets. The regions on the plot with the highest density of dots are the so–called allowed regions of the Ramachandran plot, also called low–energy regions. Some values of ϕ and ψ are forbidden since some atoms will come too close to each other, resulting in a so–called "steric clash". We know that when two atoms are too close to each other the energy of the system gets too high. For a high–quality experimental structure these regions are usually empty or almost empty –very few amino acid residues in proteins have their torsion angles within these regions. But there are exclusions from this rule –sometimes such values can be found and they most probably will result in some strain in the polypeptide chain. In such cases additional interactions will be present to stabilize such structures. They may have functional significance and may be conserved within a protein family [8].

Secondary Structure

The primary sequence or main chain of the protein must organize itself to form a compact structure. This is done in an elegant fashion by forming secondary structure elements (SSE). The two most common secondary structure elements (SSE) are alpha helices and beta sheets, formed by repeating amino acids with the same torsion (ϕ, ψ) angles. There are other secondary structure elements such as turns, coils, 3_{10}–helix etc. When looking at the helix in the figure 2.9, notice how the carbonyl oxygen atoms $C = O$ point in one direction, towards the amide NH groups, 4 residues away $(i, i+4)$ in the helix. Together these groups form a hydrogen bond, one of the main forces of secondary structure stabilization in proteins. Hydrogen bonds are shown on the right in figure 2.9 as dashed lines. For a hydrogen bond to be formed, two electronegative atoms (in the case of an alpha–helix the amide N, and the carbonyl O) have to interact

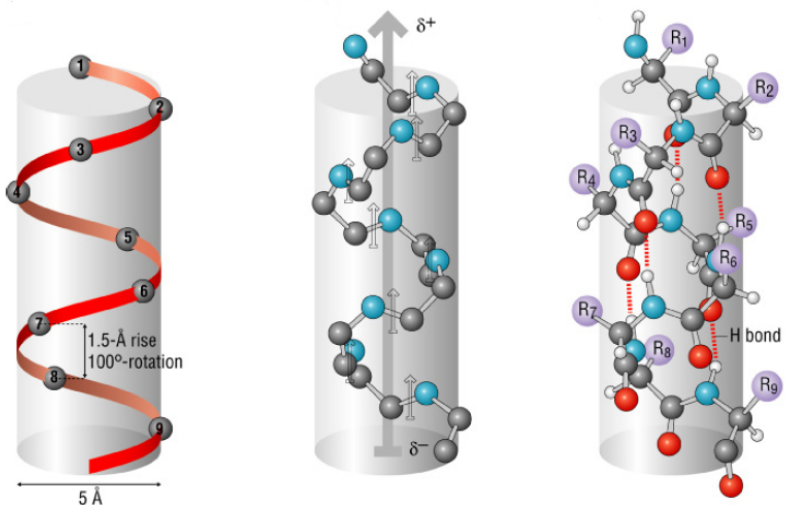


Figure 2.9: Protein Alpha Helix

with the same hydrogen. The hydrogen is covalently attached to one of the atoms (called the hydrogen–bond donor), but interacts electrostatically with the other (the hydrogen bond acceptor, O). In proteins essentially all groups capable of forming $H-$bonds (both main chain and side chain, independently of whether the residues is within a secondary structure or some other type of structure) are usually $H-$bonded to each–other or to water molecules. Due to their electronic structure, water molecules may accept 2 hydrogen bonds, and donate 2, thus being simultaneously engaged in a total of 4 hydrogen bonds. Water molecules may also be involved in the stabilization of protein structure by making hydrogen bonds with the main chain and side chain groups in proteins and even linking different protein groups together. In addition, water is often found to be involved in ligand binding to proteins, mediating ligand interactions with protein polar groups. It is useful to remember that the energy of a hydrogen bond, depending on the distance between the donor and the acceptor and the angle between them, is in the range of $2 - 10$ kcal/mol. Other types of helices in proteins include the 3_{10} helix, which is stabilized by hydrogen bonds of the type $(i, i+3)$ and the p–helix, which is stabilized by hydrogen bonds of the type $(i, i + 5)$. The 3_{10} helix has a smaller radius, compared to the alpha–helix, while the pi–helix is wider. Hydrogen bonds also stabilize another type of secondary structure in proteins, namely beta–sheets. An example of a beta–sheet with the stabilizing hydrogen bonds shown as dashed lines is presented on the figure 2.10. From the figure 2.10, the hydrogen bonds link together different segments of the protein structure. By other words, they are not formed between adjacent residues, as in alpha–helices. Rather, different segments of the

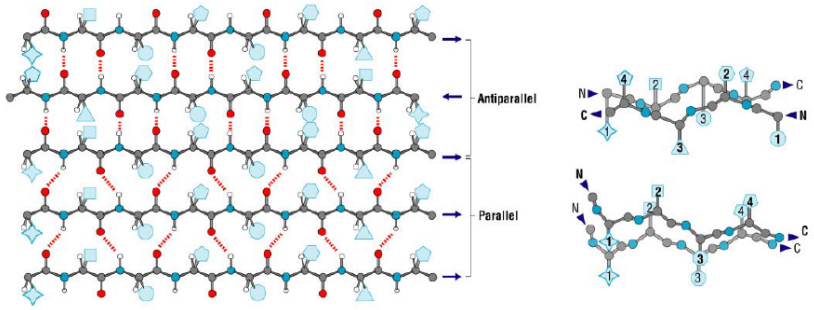
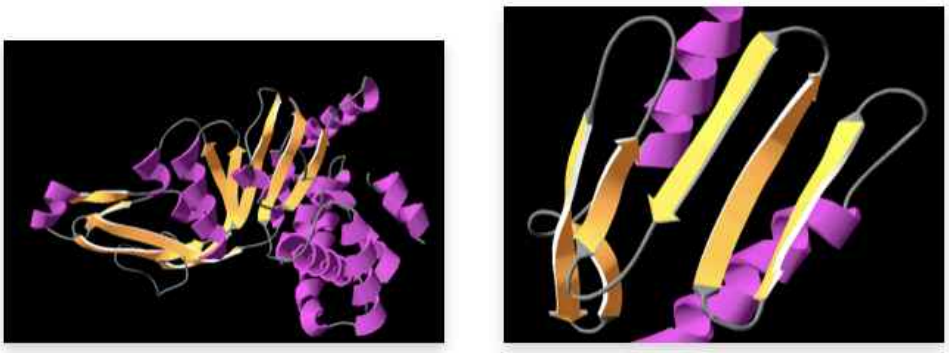


Figure 2.10: Protein Beta Sheet



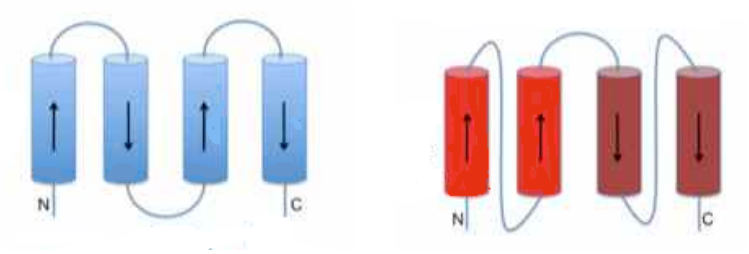
(a) Parallel beta sheet (b) Anti–parallel beta sheet

Figure 2.11: 3D Beta Sheet in ribbon representation

amino acid sequence, called beta–strands) come together to form a beta–sheet. Thus, a beta–sheet consists of several beta–strands, kept together by a network of hydrogen bonds. The same beta–sheet is shown on the figure 2.11, this time in the context of the 3D structure to which it belongs and in a so–called "ribbon" representation (protein colored according to secondary structure –beta–sheets in yellow and helices in magenta). The arrows show the direction of the beta–sheet, which is from the $N-$ to the $C-$terminus. When the arrows point in the same direction, we call such a beta–sheet parallel and when they point in opposite directions, the beta–sheet is anti–parallel.

2.2.3 Secondary Structure Element Interaction Network (SSE-IN)

After secondary structure the next level of protein structure can be described as the structural motif level, also called super–secondary structure of tertiary structure, which



(a) Anti-Parallel 4-helix bundle

(b) 4-helix bundle with 2 anti-parallel and 2 parallel helices

Figure 2.12: Alpha Helix Bundle

show the connectivity between the secondary structure elements. As mentioned before, in protein structure helices and strands are connected to each other and combined in many different ways. From known protein three dimensional structures we have learned that there is a limited number of possible ways in which secondary structure elements are combined in nature.

These secondary structure elements (SSE) connect with each other with different bonds to create tertiary structure of protein. This creates the interaction network of secondary structure element which is called SSE-IN. Probably the simplest protein structural motif can be seen in a helical bundle, shown on the schematic view in figure 2.12. Helix bundles are very common in protein structures and are very often found as separate domains within larger, multi–domain protein molecules. Another common connectivity type may be found in a parallel beta–sheet. In this case connections between the strands do not need to be of the type "short loops". When a segment of a structure is not ordered in any secondary structure type, the connectivity is called coiled regions. However, connectivity between strands in a parallel beta–sheet may also be provided by helices, building the so–called helix–strand–helix motif. In the example below both alpha–helices and coiled regions connect the strands in a parallel beta–sheet (figure 2.13).

2.2.4 Protein Folding and Classification

Protein fold assignment will often reveal evolutionary relationships, which sometimes are difficult to detect at sequence level, it helps in better understanding of protein function, its biological activity and role in living organisms. The relationship between amino acid sequence and protein three dimensional structure is not unique: different sequences, sometimes totally unrelated sequences may have similar 3D structure. By

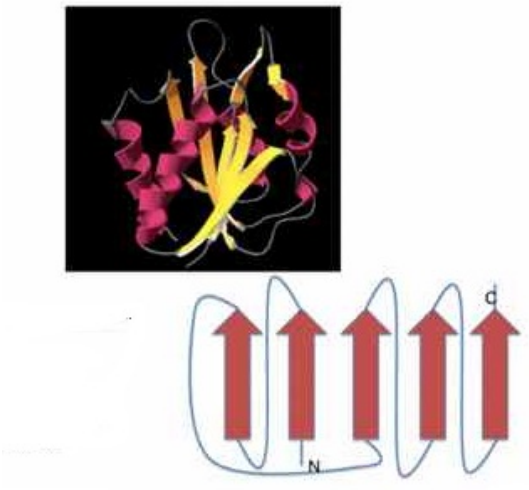
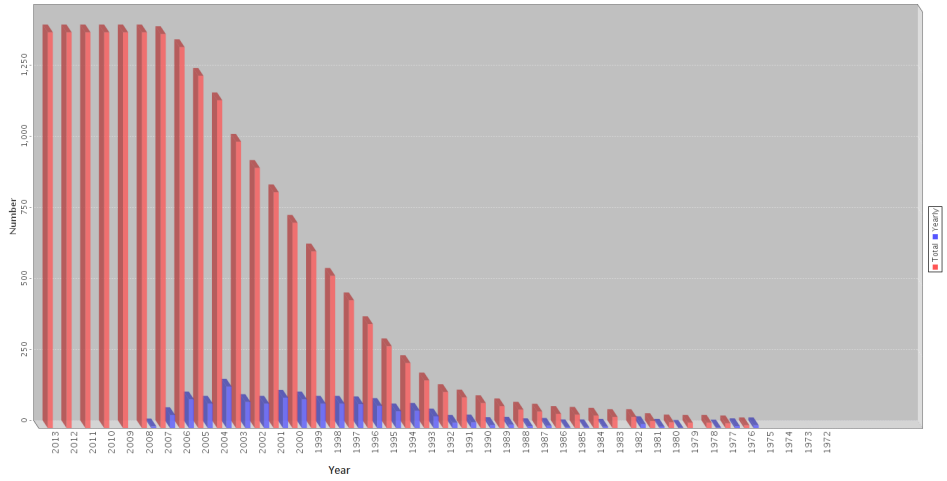


Figure 2.13: Schematic representation of the parallel $\beta-$sheet (yellow on the structural figure) in flavodoxin. Helices have been omitted form this schematic picture

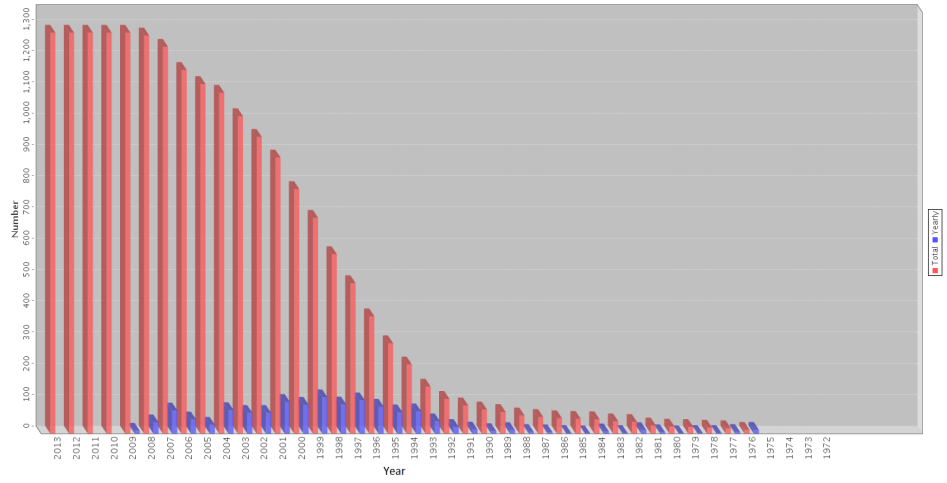
other words, the degree of conservation of the three–dimensional structure is much higher than the degree of conservation of the amino acid sequence. Three–dimensional structures sometimes may differ substantially from each other, at the sequence and even and at the structural level, but still have the same type of fold. The protein fold can be defined simply, a certain way of arrangement of secondary structure elements in space. As the number of amino acid sequence is huge, one would expect a high number of different folds. But in reality it is not like that. The number of folds is limited. Nature has re–used the same folding types again and again for performing totally new functions. Some people would refer to the common ancestor, from which all other organisms have originated. $SCOP$ and $CATH$ are the two databases generally accepted as the two main authorities in the world of fold classification. According to SCOP there are 1393 different folds. Also notice the graph in figure 2.14a, the last time a new fold was identified was 2008. The next graph in figure 2.14b shows the folds identified by CATH database, a total of 1282 folds. Apparently the two databases use slightly different fold definitions and protein fold classification, which results in different total numbers of protein folds. It is also interesting to note that during the recent years essentially no new folds have emerged.

2.2.5 Protein Tertiary Structure

Folded protein bind together to form dimer, trimer or higher order structures. These are called tertiary structure of protein. In figure 2.15, there is a tertiary structure of



(a) SCOP



(b) CATH

Figure 2.14: Yearly Growth of Total Structure

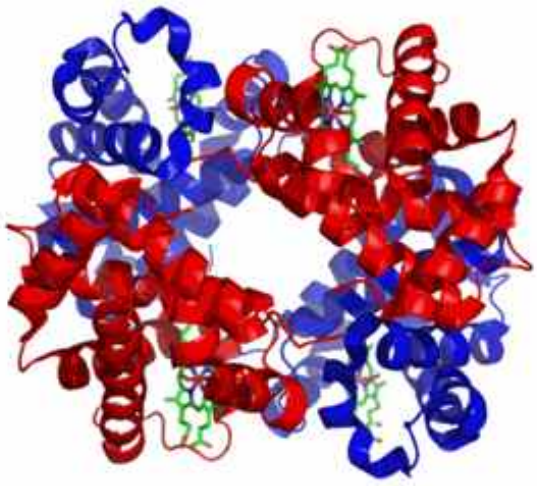


Figure 2.15: Tertiary Structure of Haemoglobin

haemoglobin protein. Tertiary structure is the final shape of protein structure hierarchy. This final shape is determined by a variety of bonding interactions between the "side chains" on the amino acids. The tertiary structure of a protein is defined by its three-dimensional structure through the atomic coordinates. Tertiary structure is formed by the packing of protein secondary structure elements into compact globular units called protein domains [9]. A whole protein might comprise one or several such domains, and its tertiary structure can refer to each individual domain as well as to the complete configuration of the whole protein, provided it contains a single, contiguous polypeptide chain backbone.

Chapter 3

Literature Review

The folding process of amino acid interaction network is related to protein structure and protein folding process. Thus in this chapter we will discuss about some state of the art literature and algorithms on protein structure prediction and protein folding algorithms as well as about some previous work on amino acid interaction network, though there are few works on amino acid interaction.

3.1 Protein Structure Prediction

A grand challenge has been to develop a computer algorithm that can predict a proteins 3D native structure from its amino acid sequence. On the one hand, knowledge of native structures is a starting point for understanding biological mechanisms and for discovering drugs that can inhibit or activate those proteins. On the other hand, we know 1000-fold more sequences than structures, and this gap is growing because of developments in high-throughput sequencing. So, there is considerable value in methods that could accurately predict structures from sequences.

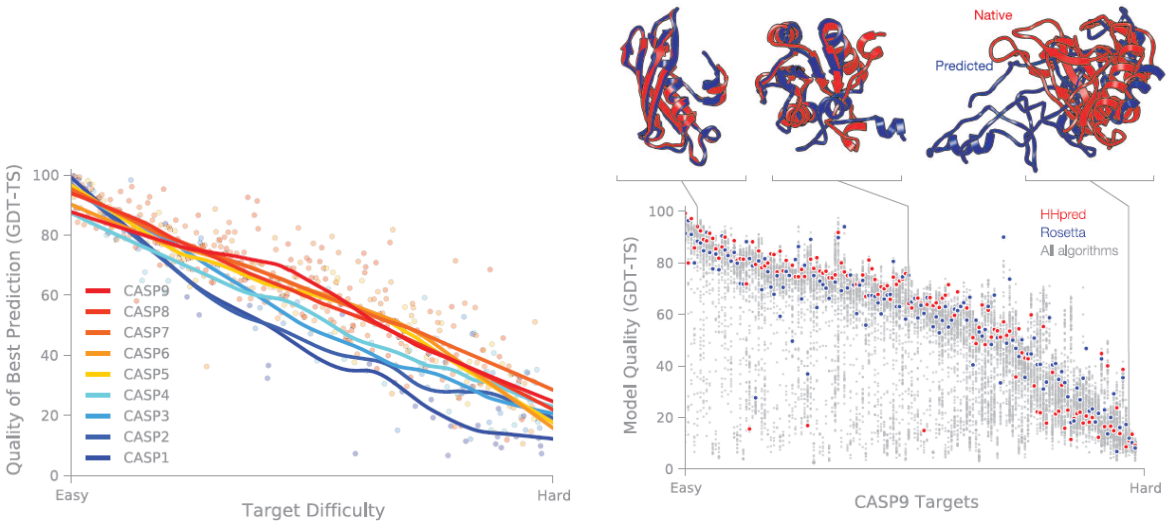
Computer-based protein-structure prediction has been advanced by Moult and colleagues, in an event initiated in 1994 called CASP: Critical Assessment of protein Structure Prediction [10; 11]. Held every second summer, CASP is a community wide blind competition in which typically more than 100 different target sequences (of proteins whose structures are known but not yet publicly available) are made available to a community that numbers more than 150 research groups around the world. Each participating group applies some algorithmic scheme that aims to predict the 3D structures of these target proteins. After each CASP event, the true experimental structures are then revealed, group performances are evaluated, and community evaluations are

published.

Currently, all successful structure-prediction algorithms are based on assuming that similar sequences lead to similar structures. These methods draw heavily on the PDB, which now contains more than 80,000 structures. However, many of these structures are similar, and the PDB contains only ∼4000 structural families and 1200 folds [12]. CASP-wide progress over the past 18 years is summarized in Fig. 3.1a. Prediction accuracies improved from CASP1 (1994) to CASP5 (2002) on the basis of several advances: (i) PDB expanded from 1600 structures to 19,000 during that time. (ii) Better sequence search and alignment tools, such as Position-Specific Iterated Basic Local Alignment Search Tool (PSI-BLAST) [13], enabled the detection of more remote evolutionary relationships and more accurate sequence alignments. (iii) A strategy, called the fragment assembly approach [14; 15; 16; 17], was developed that can often improve predictions when a similar sequence cannot be found in the PDB. If the target proteins sequence is related to a sequence that is already in the PDB, predicting its structure is usually easy (Fig. 3.1). In such cases, target protein structures are predicted by using template-based modelling (also called homology modelling or comparative modelling). But when there is no protein in the PDB with a sequence resembling the targets, accurately predicting the structure of the target is much more difficult. These latter predictions are called free modelling (also called ab initio or de novo prediction). One of the most successful free-modelling techniques is fragment assembly, described below.

In fragment assembly [14; 15; 16; 17], a target protein sequence is de-constructed into small, overlapping fragments. A search of the PDB is performed to identify known structures of similar fragment sequences, which are then assembled into a full length prediction. The qualities of fragments and their assemblies are assessed by using some form of scoring function that aims to select more native-like protein structures from among the many possible combinations. Problems of folding physics described above share more commonality with free modeling than with template-based modelling.

Since CASP6, although overall progress has slowed (Fig. 3.1a), there has been systematic, incremental progress [19]. The best groups can now on average produce models that are better than the single best template from the PDB. Progress has been made toward successfully combining multiple templates into a single prediction. Substantial improvements have been observed for free-modelling targets shorter than 100 amino acids, although no single group yet consistently produces accurate models. Larger free-modeling targets remain challenging. Several recent algorithmic develop-



(a) A. Historical CASP Performance

(b) B. Performance in CASP9

Figure 3.1: Historical and present performance in CASP. Model quality is judged by using GDT-TS [18], which is approximately the percentage of residues that are located in the correct position. (A) Evolution of accuracy over the history of CASP, spanning 18 years. Each target is classified according to an approximate measure of difficulty that incorporates both the structural and sequence similarity to proteins of known structure [19]. Each dot represents the best prediction (across all participants) for a given target. (B) Summary of prediction accuracy in CASP9 [20]. We highlight the performance of two of the best automated server algorithms. Selected predictions are superimposed on the corresponding native structures to give a visual sense of the accuracy level that can be expected.

ments–to predict residue–residue contacts from sequence alone [21; 22; 23] and to more sensitively and accurately identify remote homologs [24]–promise to further improve prediction accuracy. The performance of two of the best fully automated server predictors during CASP9 [20] are shown in Fig. 3.1b: HHPred, a pure template–based modeling tool [25], and ROSETTA, a hybrid tool that combines fragment assembly and template based modelling with all-atom refinement [26]. For some fraction of CASP targets [∼10%, based on a cut-off of 85GlobalDistance TestTotal Score (GDT-TS) [27], which is defined in Fig. 3.1 , legend], the best predictions are now accurate enough to interpret biological mechanisms, to guide biochemical studies, or to initiate a drug discovery program (which requires structural errors of less than 2 to 3 Å). However, it remains a challenge to predict the other 90% of protein structures this accurately. In addition, it is also critical to improve physics-based technologies and to reduce our de-

pendence on knowledge of existing structures, so that we can ultimately study protein motions, intrinsically disordered proteins, induced-fit binding of drugs, and membrane proteins and fold-able polymers, for which databases are too limited.

3.2 Protein Folding

Protein folding is a quintessential basic science. There has been no specific commercial target, yet the collateral pay-off's have been broad and deep. Specific technical advances are reviewed elsewhere [28]; below, we describe a few general outgrowths.

3.2.1 Growth of protein–structure databases

Today, more than 80,000 protein structures are known at atomic detail and publicly available through the PDB. New structures are being added at a rapid pace, supported by the National Institutes of Health (NIH)–funded Protein Structure Initiative, which was developed in part to inform protein structure prediction.

3.2.2 Advances in computing technology

Understanding protein folding was a key motivation for IBMs development of the Blue Gene supercomputer [29], now also used to study the brain, materials, weather patterns, and quantum and nuclear physics. Protein folding has also driven key advances in distributed–grid computing, such as in Folding@home, developed by Pande at Stanford, in which computer users all over the world donate their idle computer time to perform physical simulations of protein systems [30]. Folding@home, which now has more than one million registered users and an average of 200,000 user–donated CPUs available at any one time, provided some of the earliest simulations showing that MD simulations can accurately predict folding rates [31]. The Anton computer from DE Shaw Research, custom designed to simulate biomolecules, gives several orders of magnitude better performance than conventional computers [32]. Advances in computer technology have led to major advances in forcefields and to more reliable atomic–level insights into biological mechanisms.

3.2.3 Improvements in bio-molecular forcefields

Computer processing power has advanced at the Moores law rate, doubling every $\sim$2 years. But equally important, forcefields have kept pace. Increased computer power

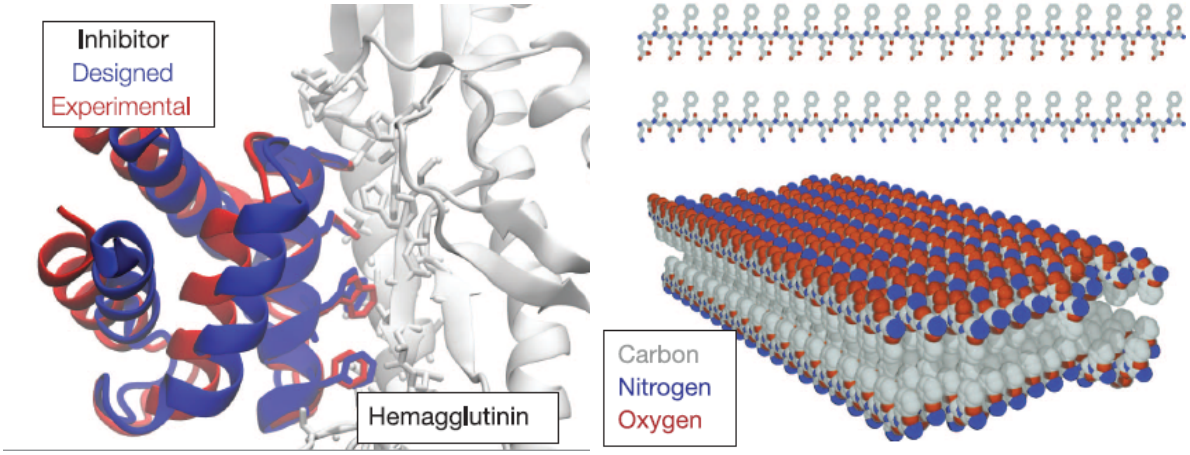
leads to longer computed time scales, which puts more stringent demands on the accuracies of bio-molecular forcefields. In a pioneering paper in 1977, McCammon *et al.* showed that the BPTI protein was stable in computer simulations during a computed time of 10 ps [33]. Today, small proteins are typically stable in explicit–water simulations for 5 to 8 orders of magnitude longer–microseconds to milliseconds of computed time [34]. Achieving such advances has required continuous improvements in forcefield accuracy.

3.2.4 New sociological structures in the scientific enterprise

Protein folding has driven innovations in how other field of science is done. CASP was among the first community-wide scientific competitions/collaborations, a paradigm for how grand–challenge science can be advanced through an organized communal effort. Other such competitions have followed, including Critical Assessment of Prediction of Interactions (CAPRI) (predicting protein-protein docking) [35],SAMPL (predicting small-molecule solvation free energies, and ligand binding modes and affinities) [36], and GPCR-Dock (predicting structures for G–protein coupled receptors, a pharmaceutically important category of membrane proteins) [37], among many others. Protein folding has also pioneered citizen science, such as in Folding [30] and Robetta@home and in a computer game called Foldit [38], in which the public engages in protein folding on their home computers.

3.2.5 New materials: Sequence-specific fold-able polymers

The principles and algorithms developed for protein folding have led to non-biological, human-made proteins and to new types of polymeric materials. In particular, proteins have been designed that bind to and inhibit other proteins (fig. 3.2a) [39], have new folds [40], have new enzymatic activities [41], and act as potential new vaccines [42]. Also, a class of non-biological polymers has emerged, called foldamers, that are intended to mimic protein structures and functions [43; 44; 45; 46]. Foldamers already have broad ranging applications [47; 48; 49; 50] as inhibitors of protein–protein interactions, broad–spectrum antibiotics, lung surfactant mimics, optical storage materials, a zinc-fingerlike binder, an RNA protein binding disrupter for application in muscular dystrophy, gene transfection agents, and molecular paper (Fig. 3.2b). Although such materials have potential applications in biomedicine and materials science, they also provide a way for us to test and deepen our understanding of protein folding.



(a) Designed Proteins (b) Designed Foldamers

Figure 3.2: Designed proteins and foldamers. (A) A protein inhibitor that was designed by computer to bind to hem-agglutinin, an influenza protein. After design, the inhibitor was crystallized in a complex with hemagglutinin. The designed structure is in remarkably good agreement with experiment, particularly for the side chains involved in binding. (B) Peptoids are synthetic, fold-able, protein–inspired polymers that have various applications. Shown here are peptoids that were designed as chains of alternating hydrophobic (gray) and either positively (blue) or negatively (red) charged side chains that spontaneously forms thin 2D structure called molecular paper.

Chapter 4

Amino Acid Interaction Network

Many systems, both natural and artificial, can be represented by networks, that is, by sites or vertices bound by links. The study of these networks is interdisciplinary because they appear in scientific fields like physics, biology, computer science or information technology. These studies are lead with the aim to explain how elements interact with each other inside the network and what are the general laws which govern the observed network properties. From physics and computer science to biology and social sciences, researchers have found that a broad variety of systems can be represented as networks, and that there is much to be learned by studying these networks. Indeed, the studies of the Web [51], of social networks [52] or of metabolic networks [53] contribute to put in light common non-trivial properties of these networks which have a priori nothing in common. The ambition is to understand how the large networks are structured, how they evolve and what are the phenomena acting on their constitution and formation.

4.1 Interaction Network General Models

In this chapter we present the three main models of interaction networks by describing their specific properties. We also define several measures that we use in the in order to study $SSE - IN$ (Secondary Structure Element Interaction Network, discussed in 2.2.3) empirically.

4.1.1 Random Graph Model

The random graph models are one of the oldest network models, introduced in [54] and further studied in [55] and [56]. These works identify two different classes of random graphs, called $G_{n,u}$ and $G_{n,p}$ and defined by the following connection rules:

- $G_{n,u}$ regroups all graphs with n vertices and m edges. To generate a graph sampled uniformly at random from the set $G_{n,u}$, one has to put m edges between vertex pairs chosen randomly from n initial unconnected vertices.

- $G_{n,p}$ is the set of all graphs consisting of n vertices, where each vertex is connected to others with independent probability p. To generate a graph sampled randomly, one has to begin with n initially unconnected vertices and join each pair by an edge with probability p.

In $G_{n,u}$ the number of edges is fixed whereas in $G_{n,p}$ the number of edges can fluctuate but its average is fixed. When n tends to be large the two models are equivalent.
Definition 1 The degree of a vertex v, k_v, is the number of edges incident to v. The mean degree, z, of a graph G is defined as follows:

$$z = \frac{1}{n}\sum_{v \in V} k_v = \frac{2m}{n} = p(n-1)$$

The degree distribution is one of the important characteristics of this kind of networks because it affects their properties and behaviour [57]. The random graph $G_{n,p}$ has a binomial degree distribution. The probability p_k that a randomly chosen vertex is connected to exactly k others is [58] :

$$p_k = \binom{n}{k} p^k (1-p)^{n-k}$$

when n is tends to infinity, this becomes:

$$p_k = \lim_{n\to\infty} \frac{n^k}{k!}\left(\frac{p}{1-p}\right)^k (1-p)^n \approx \frac{z^k e^{-z}}{k!}$$

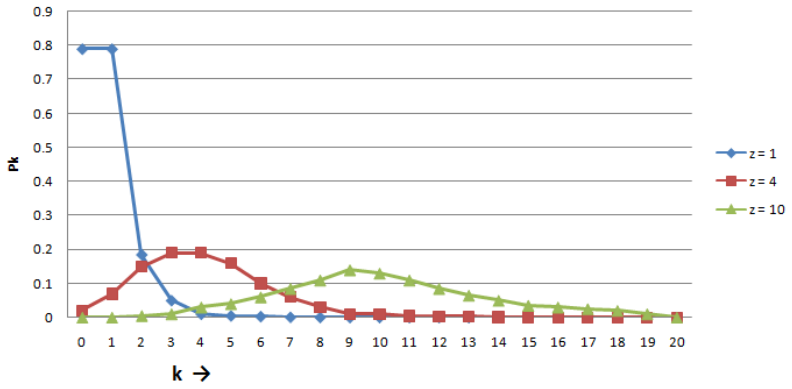


Figure 4.1: Poisson distribution $p_k = \frac{z^k e^{-z}}{k!}$ with $z = 1, 2$ and 4.

As we see in Figure 4.1, Poisson distribution have different behaviour for different mean degree z. Each distribution has a clear peak close to $k = z$, followed by a tail that decays as $1/k!$ which is considerably quicker than exponential.

4.1.2 Small-World Networks

This network model was introduced in [59] as a model of social networks. It has been since adopted to treat phenomena in physics, computer science or social sciences. The model comes from the observation that many real-world networks have the following two properties:

- The small-world effect, meaning that most pairs of vertices are connected by a short path through the network. This phenomenon has two explanations. First, the concept of "shortcuts" through a network allows to join two distant vertices by a small number of edges [59]. Second, the concept of "hubs", vertices whose connectivity is higher than others provide bridges between distant vertices because most vertices are linked to them.

- High "clustering", meaning that there is a high probability that two vertices are connected one to another if they share the same neighbour.

To determine if a network is a small-world, one can use the measures described below and compare them to the corresponding measures of a random graph.

Definition 2 The characteristic path length [59], denoted L, of a graph G is the median of the means of the shortest path lengths connecting each vertex v to all other vertices. More precisely, let $d(v, u)$ be the length of the shortest path between two vertices v and u and let $\overline{d_v}$ be the average of $d(u, v)$ over all $u \in V$. Then the characteristics path length is the median of $\{\overline{d_v}\}$

This definition applies when the graph consists of single connected component. However, the $SSE - IN$ we consider in the next section may have several connected components. In this case, when we calculate the mean of the shortest path lengths $\overline{d(v)}$ we take into account only the vertices u which are in the same connected component as v.

Since the mean and the median are practically identical for any reasonably symmetric distribution, the characteristic path length of a random graph is the mean value of the shortest path lengths between any two vertices. The characteristic path length of a random graph with mean degree z is

$$L_{RG} = \frac{logn}{logz}$$

It increases only logarithmically with the size of the network and remains therefore small even for large systems.

Definition 3 The local clustering coefficient [59], C_v, of a vertex v with k_v neighbours measures the density of the links in the neighbourhood of v.

$$C_v = \frac{|E(\Gamma_v)|}{\binom{k_v}{2}}$$

where the numerator is the number of edges in the neighbourhood of v and the denominator is the number of all possible edges in this neighbourhood. The clustering coefficient C of a graph is the average of the local clustering coefficients of all vertices:

$$C = \frac{1}{n} \sum_{v \in V} C_v$$

The clustering coefficient of a random graph with mean degree z is

$$C_{RG} = \frac{z}{n-1}$$

Watts and Strogatz [59] define a network to be a small-world if it shows both of the following properties:

1. Small world effect: $L \approx L_{RG}$
2. High clustering: $C >> C_{RG}$

4.1.3 Scale-Free Networks

The most important property of scale-free systems is their invariance to changes in scale. The term "scale-free" refers to a system defined by a functional form $f(x)$ that remains unchanged within a multiplicative factor under rescaling of the independent variable x. Indeed, this means power-law forms, since these are the only solutions to $f(an) = bf(n)$, where n is the number of vertices [60]. The scale-invariance property means that any part of the scale-free network is stochastically similar to the whole network and parameters are assumed to be independent of the system size [53].

If n_k is the number of vertices having the degree k, we define p_k as the fraction of vertices that have degree k in the network:

$$p_k = \frac{n_k}{n}$$

The degree distribution can be expressed via the cumulative degree function [60; 55]:

$$P_k = \sum_{k'=k}^{\infty} p_{k'}$$

which is the probability for a node to have a degree greater or equal to k.

By plotting the cumulative degree function one can observe how its tail evolves, following a power law or an exponential distribution.

The power law distribution is defined as following [60]:

$$P_k \approx \sum_{k'=k}^{\infty} k'^{\alpha} \approx k^{-(\alpha-1)}$$

and the exponential distribution is defined by the next formula:

$$P_k \approx \sum_{k'=k}^{\infty} e^{-k'/\alpha} \approx e^{-k/\alpha}$$

Between this two distributions, there is a mixture of them where the distribution has a power law regime followed by a sharp cut-off, with an exponential decay of the tail, expressed by the next formula:

$$P_k \approx \sum_{k'=k}^{\infty} k'^{-\alpha} e^{-k'/\alpha} \approx k^{\alpha-1} e^{-k/\alpha}$$

Like a power law distribution, it decreases polynomially, so that the number of vertices with weak degree is important while a reduced proportion of vertices having high degree exists. The last are called "hubs" that is sites with large connectivity through the network, as Figure 4.2. The scale-free model depends mainly on the kind

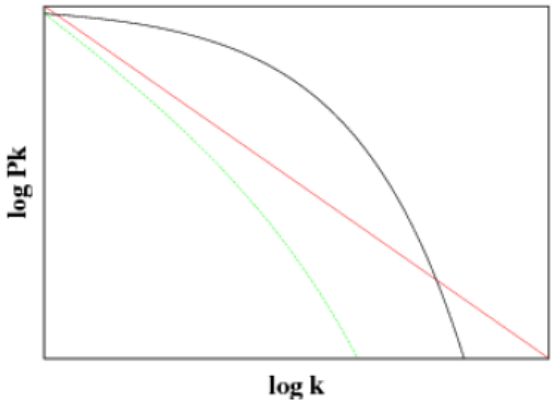


Figure 4.2: Degree distribution described in [61]. The red line follows a power law, as for scale-free networks. The green line corresponds to truncated scale-free networks. The black curve corresponds to single-scale networks.

of degree distribution, thus a network is defined as a scale-free if:

- The degree distribution is a power law distribution $P(k) \approx k^{-\alpha}$ over a part of its range.
- The distribution exposant satisfies $2 < \alpha \leq 3$ [62].

Amaral *et al.* [61] have studied networks whose cumulative degree distribution shape lets appear three kinds of networks. First, scale-free networks whose distribution decays as a power law with an exposant α satisfying bounds seen above. Second, as Figure 4.2, broad-scale or truncated scale-free networks whose the degree distribution has a power law regime followed by a sharp cut off. Third, single-scale networks whose degree distribution decays fast like an exponential.

4.1.4 Topological Measures

Here, we present some measures that we use to describe proteins' $SSE - IN$. Among them, there are simple ones, the most frequently used, but also more subtle, which allow a more precise discrimination between interaction networks.

Diameter and mean distance The distance in a graph $G = (V, E)$ between two vertices $u, v \in V$, denoted by $d(u, v)$, is the length of the shortest path connecting u and v [63]. If there is no path between u and v, we suppose that $d(u, v)$ is undefined. A graph diameter, D, is the longest shortest path between any two vertices of a graph [63]:

$$D = max\{d(u, v)|u, v \in V\}$$

The mean distance is defined as the average distance between each couple of vertices:

$$\overline{d_G} = \frac{2}{n(n-1)} \sum_{u,v \in V} d(u, v)$$

Density The density, denoted δ, is defined as the ratio between the number of edges in a graph and the maximum number of edges which it could have:

$$\delta(G) = \frac{2m}{n(n-1)} \approx \frac{2m}{n^2}$$

The density of a graph is a number between 0 and 1. When the density is close to one, the graph is called dense, when it is close to 0, the graph is called sparse [64].

Clustering coefficients Watts and Strogatz proposed a measure of clustering [59] and defined it as a measure of local vertices density, thus for each node v, the local clustering around its neighbourhood is defined in the following way:

$$C_v = \frac{1}{2}k_v(k_v - 1)$$

The clustering coefficient is a ratio between the number of edges and the maximum number of possible edges in the vertices neighbourhood. If we extend the previous definition to the entire graph, the clustering is given by the expression:

$$C_{local} = \frac{1}{n}\sum_{v \in V}\frac{N_{connected}}{C_v}$$

where, $N_{connected}$ is number of connected neighbour pairs.

The last definition is mainly local because for each node, it involves only its neighbourhood. The global clustering was studied by Newman *et al.* [58] and can be measured by the following formula:

$$C_{global} = \frac{3 \times N_{triangle}}{N_{triplet}}$$

where, $N_{triangle}$ is number of triangle in the graph and $N_{triplet}$ is number of connected triplets of vertices.

A triangle is formed by three vertices which are all connected and a triplet is constituted by three nodes and two edges. The global clustering coefficient C_{global} is the mean probability that two vertices that are neighbours of the same other vertex will themselves be neighbours.

4.2 Topological Description

The behaviour or SSE-IN is studied in [65] published by Gaci *et al.*,which we are going to describe in this section. We want to observe how proteins from a same structural family provide similar SSE-IN according to their topological properties. To do that, we propose topological measures which we apply on a sample of proteins to put in evidence the existence of equivalence between structural similarity and topological homogeneity in the resulting SSE-IN.

The first step before studying the proteins SSE-IN is to select them according to their SSE arrangements. Thus, a protein belongs to a CATH topology level or a SCOP

Name	Class	Proteins
RossmannFold	$\alpha\ \beta$	2576
TIM Barrel	$\alpha\ \beta$	1051
Lysozyme	Mainly α	871

Table 4.1: CATH type studied protein family

Name	Class	Proteins
Globin-like	All α	733
TIM β / α -barrel	α / β	896
Lysozyme-like	α+β	819

Table 4.2: SCOP type studied protein family

fold level iff all its domains are the same. We have worked with the CATH v3.1.0 and SCOP 1.7.1 files. We have computed the measures from the previous section for three families of each hierarchical classification, namely SCOP and CATH as in Table. 4.1 and Table. 4.2. We have chosen these three families by classification, in particular because of their huge protein number. Thus, each family provides a broad sample guaranteed more general results and avoiding fluctuations. Moreover, these six families contain proteins of very different sizes, varying from several dozens to several thousands amino acids in SSE.

4.2.1 Diameter and mean distance

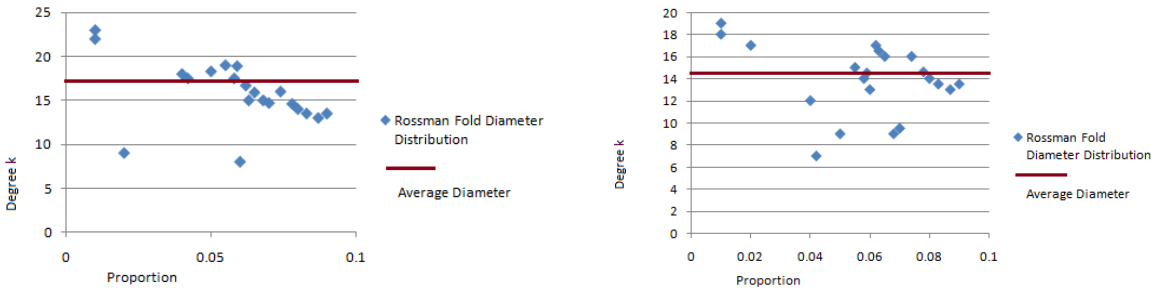
As we can see the average diameter for each 1 of the studied family in Table. 4.3 and 4.4, we can observe very close diameters between $TIMBarrel$ and $TIM\beta/\alpha - barrel$ and also between $Lysozyme$ and $Lysozyme-like$ families. This is explained by the fact that each pair of families contain almost the same proteins, in other words, $Lysozyme$ topology in CATH is the equivalent of $Lysozyme-like$ fold level in SCOP. If we closely

Protein Family Name	Diameter
RossmannFold	18.84
TIM Barrel	19.83
Lysozyme	12.81

Table 4.3: Average diameter for CATH type studied protein family

Protein Family Name	Diameter
Globin-like	15.65
TIM β / α -barrel	20.09
Lysozyme-like	12.85

Table 4.4: Average diameter for SCOP type studied protein family



(a) Average Diameter of Rossman fold

(b) Average Diameter of β/α-barrel

Figure 4.3: Average Diameter Distribution

observe the Figure. 4.3, where distribution of diameter of the two studied families are depicted, the distribution follows roughly a Poison law. These result confirm that the mean diameter is suitable property to discriminate families between them.

The diameter being an upper bound of distances in interaction networks, we expect that the mean distance z will be lower than D. Table. 4.5 confirms this. Again, we observe very close values between the equivalent SCOP and CATH families for the reasons discussed above. But we can also see that different families have values which allow discrimination between them based on this parameter. It is interesting to note that the ratio D/z is about 2.5 for all the families. The last property is a characterization of all proteins' $SSE - IN$.

4.2.2 Density and mean degree

As defined earlier, the density measures the ratio between the number of available edges and the number of all possible edges. Results presented in Table 4.6 show that the two families $TIMBarrel$ and $TIM\beta/\alpha - barrel$ have the minimum density. It has a consequence on their $SSE - IN$ topology. When the density is low, the network is less connected and consequently, the diameter and the average distance are higher.

Protein Family Name	$\overline{d_G}$
RossmannFold	7.26
TIM Barrel	7.79
Lysozyme	4.99
Globin-like	6.64
TIM β / α -barrel	7.86
Lysozyme-like	5.03

Table 4.5: Average of mean distances for each family

Protein Family Name	$\delta(G)$
RossmannFold	0.033
TIM Barrel	0.03
Lysozyme	0.038
Globin-like	0.034
TIM β / α -barrel	0.029
Lysozyme-like	0.042

Table 4.6: Average of density for each family

Comparing these results to Tables 4.4 and 4.5 one can see the inversely proportional relation between density in one hand, and diameter and average distance on the other.
The mean degree, z, is presented in Table 4.7. The observed values are close enough from one family to another. That is why the mean degree is not discriminating property, but rather a property characterizing all proteins' SSE-IN.

4.2.3 Degree Distribution

We compute the cumulative degree distribution for all proteins SSE-IN of studied families. A sample of our results is presented on Figure. 4.4. We can remark that the curves follow a power law distribution and can be approximated by the following power-law function:

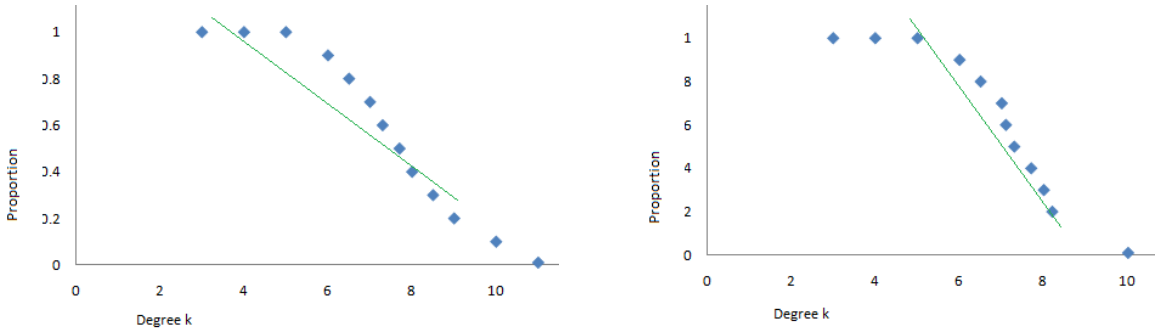
$$p_k = 141.29k^{-\alpha}$$

where $\alpha \approx 2.99$.

We observe the same results for all studied proteins. To explain this phenomenon, we have to rely on two facts. First, the mean degree of all proteins SSE-IN is nearly

Protein Family Name	z
RossmannFold	7.2
TIM Barrel	7.12
Lysozyme	6.82
Globin-like	7.69
TIM β / α -barrel	7.15
Lysozyme-like	6.81

Table 4.7: Average of mean degree for each family



(a) 1RXC cumulative distribution

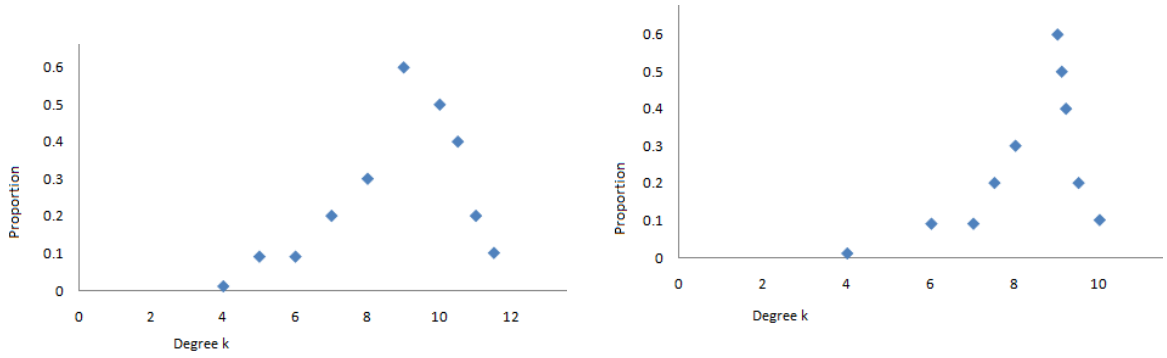
(b) 1HV4 cumulative distribution

Figure 4.4: Cumulative degree distribution for a) 1RXC from Rossman fold and b) 1HV4 from TIM β/α-barrel

constant (Table. 4.7). Second, the degree distribution, see Figure 4.5, follows a Poisson distribution whose peak is reached for a degree near z. These two facts imply that for degree lower than the peak the cumulative degree distribution decreases slowly and after the peak its decrease is fast compared to an exponential one. Consequently, all proteins SSE-IN studied have a similar cumulative degree distribution which can be approximated by a unique power-law function.

4.2.4 Clustering Coefficients

The local clustering C_{local} measures the fraction of pairs of a vertex's neighbours and the global clustering C_{global} gives the probability that among three vertices at least two are connected. The results presented in Table 4.8 show that the clustering coefficients are close for different families and cannot be correlated to density values. Consequently,



(a) 1RXC degree distribution

(b) 1HV4 degree distribution

Figure 4.5: Degree distribution for a) 1RXC from Rossman fold and b) 1HV4 from TIM β/α-barrel

Protein Family Name	C_{local}	C_{global}
RossmannFold	0.63	0.56
TIM Barrel	0.64	0.57
Lysozyme	0.65	0.58
Globin-like	0.63	0.57
TIM β / α -barrel	0.64	0.57
Lysozyme-like	0.66	0.58

Table 4.8: Clustering coefficients for each family

the neighbour density remains independent of the previously studied properties.

4.2.5 Consequences of the discussion

In this part we introduce the notion of interaction network of amino acids of a protein (SSE-IN) and study some of the properties of these networks. We give different means to describe a protein structural family by characterizing their SSE-IN. Some of the properties, like diameter and density, allow discriminating two distinct families, while others, like mean degree and power law degree distribution, are general properties of all SSE-IN. Thus, proteins having similar structural properties and biological functions will also have similar SSE-IN properties. In this way our model allows us to draw a parallel between biology and graph theory.

4.3 Amino Acid Interaction Network

Though the term amino acid interaction network not so well known in the field of proteomics, it is now a good model to research in protein folding. The 3D structure of a protein is determined by the coordinates of its atoms. This information is available in Protein Data Bank (PDB) [6], which regroups all experimentally solved protein structures. Using the coordinates of two atoms, one can compute the distance between them. We define the distance between two amino acids as the distance between their $C\alpha$ atoms. Considering the $C\alpha$ atom as a center of the amino acid is an approximation, but it works well enough for our purposes. Let us denote by N the number of amino acids in the protein. A contact map matrix is a $N \times N$ $0-1$ matrix, whose element (i, j) is one if there is a contact between amino acids i and j and *zero* otherwise. It provides useful information about the protein. For example, the secondary structure elements can be identified using this matrix. Indeed, $\alpha-$helices spread along the main diagonal, while $\beta-$sheets appear as bands parallel or perpendicular to the main diagonal [66]. There are different ways to define the contact between two amino acids. In [67], the notion is based on spacial proximity, so that the contact map can consider non–covalent interactions. Gaci *et al.* says that two amino acids are in contact iff the distance between them is below a given threshold. A commonly used threshold is 7 Å. Consider a graph with N vertices (each vertex corresponds to an amino acid) and the contact map matrix as incidence matrix. It is called contact map graph. The contact map graph is an abstract description of the protein structure taking into account only the interactions between the amino acids. Now let us consider the subgraph induced by the set of amino acids participating in SSE. We call this graph SSE interaction network ($SSE - IN$) and this is the object we study in the present chapter. The reason of ignoring the amino acids not participating in SSE is simple. Evolution tends to preserve the structural core of proteins composed from SSE. In the other hand, the loops (regions between SSE) are not so important to the structure and hence, are subject to more mutations. That is why homologous proteins tend to have relatively preserved structural cores and variable loop regions. Thus, the structure determining interactions are those between amino acids belonging to the same SSE on local level and between different SSEs on global level. In [68] and [69] the authors rely on similar models of amino acid interaction networks to study some of their properties, in particular concerning the role played by certain nodes or comparing the graph to general interaction networks models. Thanks to this point of view the protein folding

problem can be tackled by graph theory approaches.

Chapter 5

Problem Formulation

To solve a problem, first we have to formulate it with respect to some objectives. After some knowledge about structural protein in background study in Chapter 2 and 4 and about some previous research in protein folding field in Chapter 3 we can formulate the amino acid interaction network prediction problem in this chapter. As we have discuss in Chapter 4, we can define a protein structure as a network or graph where amino acid atoms are vertices of the graph and different type of bond and function or in a simple word, interaction, between two amino acids in a protein structure are edges of the graph. In a protein structure, some amino acids highly interacts with each other and forms a group or strong network, called Secondary Structure Element (SSE), and this interaction network is called SSE-IN. A protein can have hundreds of SSE-IN. Amino acids in a SSE-IN is loosely connected with other SSE-IN in a protein. In this research we are going to predict the network and its interactions from a associated known or similar protein family in Protein Data Bank.

From the above discussion in terms of mathematics and graph theory, we can define the problem as prediction of a graph $\mathcal{G}$ consist of $\mathcal{N}$ vertices $\mathcal{V}$ and $\mathcal{E}$ edges. If two amino acids interact with each other in protein we mention it as a edge $(u, v) \in \mathcal{E}, u \in \mathcal{V}, v \in \mathcal{V}$ of the graph. A SSE-IN is a highly dense sub-graph $\mathcal{G}_{SSE-IN}$ with edge set $\mathcal{E}_{SSE-IN}$. Probability of the edge $(u, v) \in \mathcal{E}_{SSE-INA}, u \in \mathcal{V}_{SSE-INA} v \in \mathcal{V}_{SSE-INA}$ is very high and probability of the edge $(u, v) \notin \mathcal{E}_{SSE-INA}, u \in \mathcal{V}_{SSE-INA} v \in \mathcal{V}_{SSE-INB}$ is very low, where $\mathcal{V}_{SSE-INA}$ and $\mathcal{V}_{SSE-INB}$ are respectively the vertex set of SSE-IN A and SSE-IN B.

In the Chapter 4, we have discussed about some properties of a network like average mean degree, average distance etc. These properties of a network are topological properties which are uses to limit the predicted network from forming pandemic one.

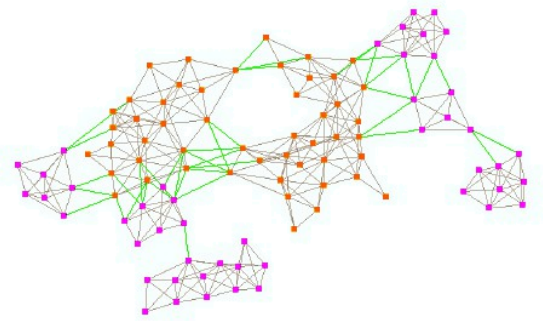


Figure 5.1: SSE-IN of 1DTP protein. Green edges are to be predicted by ant colony algorithm

To predict the network we have to solve three problems,

1. Find a associate SCOP protein family from the given protein sequence.
2. Predict a network of amino acid secondary structure element (SSE) from the known SCOP protein family.
3. Predict interactions between amino acids in the network, including internal edges of SSE-IN and external edges.

We can formulate these problem respectively as,

1. Topological coordination of graph.
2. Multi-objective optimization using Genetic Algorithm.
3. Ant Colony Optimization.

Before proposing a algorithm to solve these problem we are to formulate these problems.

5.1 Topological property

We have discussed about topological measurements of a network in section 4.1.4. From these discussion we can infer that distance between vertices in a secondary structure element (SSE) should maintain almost same property like average mean degree, average mean distance between vertices. On the other hand vertices between two different SSE should show different property. From these topological properties of a network we can define how many similar sub-graph like SSE have in the network or in the protein network. From these measurement we can find a associate SCOP family in PDB which has same number of SSE and which should be the first assumption of the network.

5.2 Multi-objective Optimization

It is very difficult to define real world problems in terms of a single objective. A multi-objective optimization problem deals with more than one objective functions that are to be minimized or maximized. These objectives can be conflicting, subject to certain constraints and often lead to choosing the best trade-off among them. Two ways of multiple-objective optimization are known, one of which is combining each individual objectives into a single composite objective. Problem lies in selecting how to combine these objectives. Various fitness functions are introduced for representing the objectives. Utility measure for those functions can be used. No one solution can be found which is best among all the others in all objectives. Therefore set of good solutions are taken.

Another way of optimizing multi-objective is determining the Pareto Optimal solution set. In a Pareto Optimal set, all solutions are non-dominated with respect to each other. We say that a solution is called non-dominated and can be considered as Pareto optimal when it is not worse in any of the objectives than the other solutions and best in at least one objective compared to others.

5.2.1 Formulation of Multi-objective Optimization

Let us define an individual X having more than one objective functions to be optimized, as an n-dimensional vector with n variables $x_1, x_2, ..., x_n$. Now if we want to make a composite objective using these single objectives, we can multiply the variables with a weight and take the weighted sum of those objectives.

The second approach is using the Pareto optimal solution set. Let us define the fitness values for each single objective as $z_1, z_2, ..., z_n$. To find the Pareto optimal set, we must find all the individuals which are non-dominated with respect to other ones. An individual X is said to dominate another individual Y and denoted as $X \prec Y$ if and only if $z_i(X) \geq z_j(Y)$ where $i, j = 0, 1, ..., k$ and k is the number of objectives and $z_i(X) > z_j(Y)$ for at least one objective function.

For a given Pareto set, the values of the respective objective functions in the problem domain are said to be Pareto front. However, identifying the actual Pareto optimal set is not feasible at all due to the size of the problem domain. Therefore, we try to find the best possible solutions. It is desirable that the best possible solutions must satisfy the following three criteria:

1. It must be close to the actual solution.

2. The solutions must be distributed over the whole problem domain.

3. It must ensure that it captures the whole range of the Pareto front and thereby provide solutions even at the extreme end.

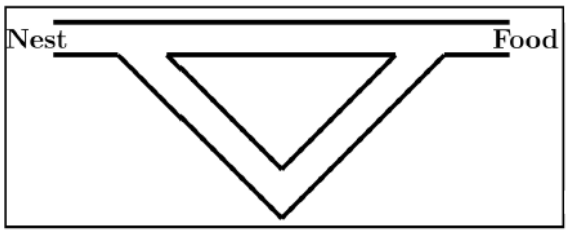
We can define the prediction of interaction network as a optimization problem. We are to find a network which is as close as possible with the associate SCOP protein family, which we have found from the topological conformation. We have three different and conflicting objectives $z_{o_1}, z_{o_2}, z_{o_3}$ for the interaction network prediction problem. These are z_{o_1} is distance between two amino acids, z_{o_2} is the relative torsion angle, a value between 0 to 1 and z_{o_3} is binary value 0 or 1, where 0 reflecting hydrophilicity and 1 is hydrophobicity of the protein SSE-IN in the network. We are to find a network from the given protein sequence, which is a optimized and as much related as possible to the associated SCOP protein family. But if we consider only the distances between amino acids, we may miss one of the most controlling property of protein the torsion angle. On the other hand, as we know the core of a protein is hydrophobic and the outer is is hydrophilic, we can assume that two amino acids with same hydrophobicity property are related. So we are to consider these three property of protein to predict the interaction network. Optimization in one objective may lead to a situation where other objectives becomes worst. So to find a optimal solution we have to make trade off between objectives.

From the above formulation about interaction network prediction of protein and multi-objective optimization, we can define the interaction network prediction problem as a multi-objective optimization problem.

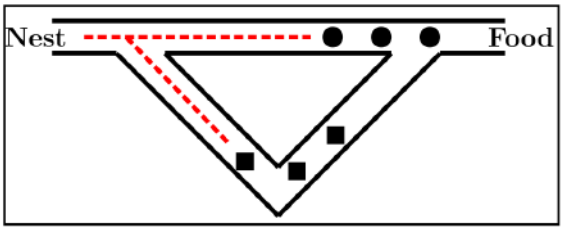
5.3 Ant Colony Optimization (ACO)

Ant colony optimization (ACO) is one of the most recent techniques for approximate optimization. The inspiring source of ACO algorithms are real ant colonies. More specifically, ACO is inspired by the ants' foraging behaviour. At the core of this behaviour is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short paths between their nest and food sources. This characteristic of real ant colonies is exploited in ACO algorithms in order to solve, for example, discrete optimization problems.

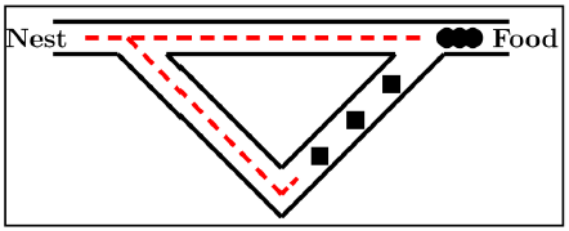
Depending on the point of view, ACO algorithms may belong to different classes of approximate algorithms. Seen from the artificial intelligence (AI) perspective, ACO



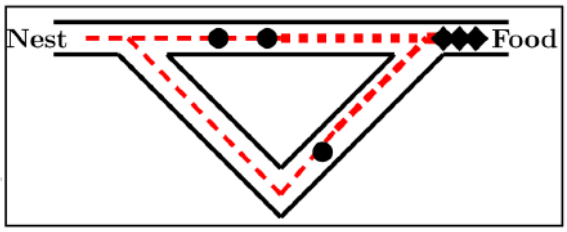
(a) All ants are in nest. There is no pheromone in the environment.



(b) The foraging starts. In probability, 50% of the ants take the shorter path (symbolized by circles), and 50% take the long path to the food source (symbolized by rhombs).



(c) The ants that have taken the shorter path have arrived earlier at the food source. Therefore, when returning, the probability to take again the shorter path is higher.



(d) The pheromone trail on the shorter path receives, in probability, a stronger reinforcement, and the probability to take this path grows. Finally, due to the evaporation of the pheromone on the long path, the whole colony will, in probability, use the shorter path.

Figure 5.2: An experimental setting that demonstrates the shortest path finding capability of ant colonies. Between the ants' nest and the only food source exist two paths of different lengths. In the four graphics, the pheromone trails are shown as dashed lines whose thickness indicates the trails' strength.

algorithms are one of the most successful strands of swarm intelligence [70; 71]. The goal of swarm intelligence is the design of intelligent multi-agent systems by taking inspiration from the collective behaviour of social insects such as ants, termites, bees, wasps, and other animal societies such as flocks of birds or fish schools. Examples of swarm intelligent algorithms other than ACO are those for clustering and data mining inspired by ants' cemetery building behaviour [72], those for dynamic task allocation inspired by the behaviour of wasp colonies [73], and particle swarm optimization [74]. Marco Dorigo and colleagues introduced the first ACO algorithms in the early 1990's [75; 76; 77]. The development of these algorithms was inspired by the observation of ant colonies. Ants are social insects. They live in colonies and their behaviour is

governed by the goal of colony survival rather than being focused on the survival of individuals. The behaviour that provided the inspiration for ACO is the ants' foraging behaviour, and in particular, how ants can find shortest paths between food sources and their nest. When searching for food, ants initially explore the area surrounding their nest in a random manner. While moving, ants leave a chemical pheromone trail on the ground. Ants can smell pheromone. When choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. It has been shown in [78] that the indirect communication between the ants via pheromone trails–known as *stigmergy*-enables them to find shortest paths between their nest and food sources. This is explained in an idealized setting in Figure 5.2.

As a first step towards an algorithm for discrete optimization we present in the following a discretized and simplified model of the phenomenon as explained in Figure 5.2. According to Blum *et al.* in [79] the model consists of a graph $G = (V, E)$, where V consists of two nodes, namely v_s (representing the nest of the ants), and v_d (representing the food source). Furthermore, E consists of two links, namely e_1 and e_2, between v_s and v_d. To e_1 we assign a length of l_1, and to e_2 a length of l_2 such that $l_2 > l_1$. In other words, e_1 represents the short path between v_s and v_d, and e_2 represents the long path. Real ants deposit pheromone on the paths on which they move. Thus, the chemical pheromone trails are modeled as follows. We introduce an artificial pheromone value τ_i for each of the two links e_i, $i = 1, 2$. Such a value indicates the strength of the pheromone trail on the corresponding path. Finally, we introduce n_a artificial ants. Each ant behaves as follows:

Starting from v_s (i.e., the nest), an ant chooses with probability

$$p_i = \frac{\tau_i}{\tau_1 + \tau_2}, i = 1, 2, \tag{5.1}$$

between path e_1 and path e_2 for reaching the food source v_d. Obviously, if $\tau_1 > \tau_2$, the probability of choosing e_1 is higher, and vice versa. For returning from v_d to v_s, an ant uses the same path as it chose to reach v_d, and it changes the artificial pheromone value associated to the used edge. More in detail, having chosen edge e_i an ant changes the artificial pheromone value τ_i as follows:

$$\tau_i \leftarrow \tau_i + \frac{Q}{l_i} \tag{5.2}$$

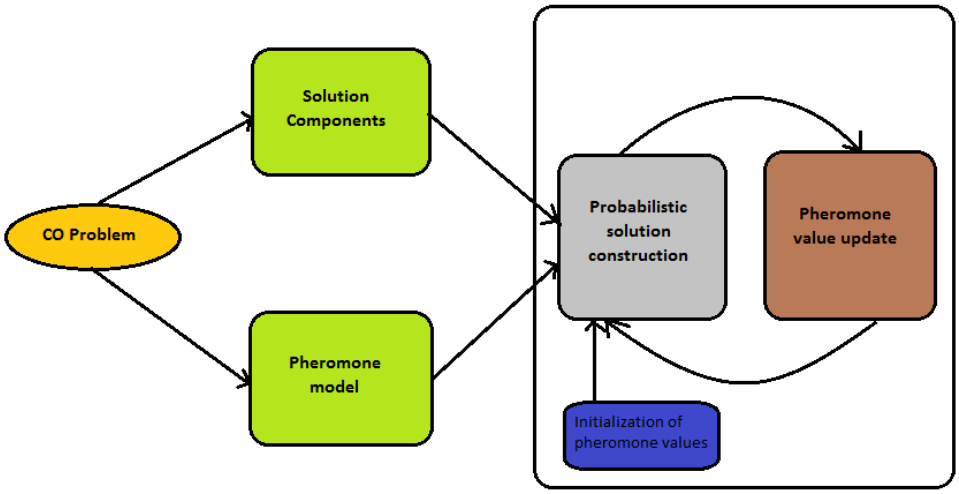


Figure 5.3: Working procedure of ACO meta-heuristic

where the positive constant Q is a parameter of the model. In other words, the amount of artificial pheromone that is added depends on the length of the chosen path: the shorter path, the higher the amount of added pheromone.

The foraging of an ant colony is in this model iteratively simulated as follows: At each step (or iteration) all the ants are initially placed in node v_s. Then, each ant moves from v_s to v_d as outlined above. As mentioned in the caption of Figure 5.2d, in nature the deposited pheromone is subject to an evaporation over time. We simulate this pheromone evaporation in the artificial model as follows:

$$\tau_i \leftarrow (1 - \rho) \cdot \tau_i, i = 1, 2. \tag{5.3}$$

The parameter $\rho \in (0, 1]$ is a parameter that regulates the pheromone evaporation. Finally, all ants conduct their return trip and reinforce their chosen path as outlined above.

ACO algorithm is basically works as shown in Figure 5.3. To solve a CO problem, to assemble solutions to the CO problem first one has to find or derive a finite set of solution components $\mathcal{S}$ and second he has to define a set of pheromone values $\mathcal{T}$. This set of values are commonly called the *pheromone model*, which is technically a parameterized probabilistic model. The ACO meta-heuristic have two major central components named solution components and pheromone model. The pheromone values $\tau_i \in \mathcal{T}$ are usually associated to solution components. The pheromone model probabilistically generate solutions to the problem under consideration by assembling

them from the set of solution components. In general, the ACO approach attempts to solve an optimization problem by iterating the following two steps according to [79]:

1. candidate solutions are constructed using a pheromone model, that is, a parameterized probability distribution over the solution space.

2. the candidate solutions are used to modify the pheromone values in a way that is deemed to bias future sampling toward high quality solutions.

In a predicted amino acid interaction network, all the interaction have to predict again to mark the edges of the network. We have to predict the inter-SSE-IN and intra-SSE-IN edges and should not distract much from the known associate SCOP protein family. As we have to maintain the three objectives $z_{o_1}, z_{o_2}, z_{o_3}$, we are to combine these objective to give weight and find paths in interaction network. Edges in these paths will be the predicted interactions of network. We have to choose paths probabilistically. Ant colony optimization is one of the best optimization techniques suited for this problem.

Chapter 6

Proposed Algorithm

As discussed in the previous chapters we can formulate a protein into model or a graph of amino acids and as we have seen about multi-objective optimization problem in section 5.2 and ant colony optimization (ACO) in section 5.3, we can solve the protein folding problem using these two new and emerging algorithms. The multi-objective optimization algorithm will predict structural motifs of a protein and will give a network or graph of secondary structural element (SSE) of the protein. On the other hand, the ant colony optimization (ACO) algorithm will find the interactions between amino acids including the intra-SSE-IN and inter-SSE-IN interactions. In our algorithm we have considered a folded protein in the PDB as an unknown sequence if it has no SCOP v1.73 family classification. According to [80], we can associate the most compatible and best fit structural family based on topological criteria like average diameter, average mean distance etc. In our research we considered an unknown sequence as protein of known amino acids sequence.

Protein folding process is oriented and proteins do not explore entire conformational space as it follows the Levinthal hypothesis also called the kinetic hypothesis. To follow the kinetic hypothesis we limited the topological space by a related structural family of the given sequence [80] to fold a SSE-IN.

6.1 Associate Protein Family

As a first step to predict a amino acid interaction network of protein, we have to choose an associate protein family, to which we are going to use as a reference structural motif. Though Protein Data Bank (PDB) contains thousands of proteins motifs to choose as a associate protein family, we can use already existing sequence alignment algorithms to

find out close sequence from the PDB. We have chosen Basic Local Alignment Search Tool (BLAST) [1] tool to find the most similar protein family from PDB.

6.2 Prediction of SSE interaction network using Multi-objective Optimization

In chapter 5 we have defined that our first problem, predicting SSE interaction network as a multi-objective optimization. There are several ways to solve multi-objective optimization problem. In this research we have decided to use Genetic Algorithm(GA) as multi-objective optimization. The GA has to predict the adjacency matrix of unknown sequence when it is represented by chromosome.

6.2.1 Genetic Algorithms as Multi-objective Optimization

Genetic algorithm (GA)is much similar to the process of natural evolution. It generates useful solutions to optimization and search problems. Genetic algorithms can be considered as the subclass of evolutionary algorithms (EA).

In genetic algorithm terminology, a solution is called individual or chromosome[81]. The discrete units that make up the chromosome is called gene. The collection of chromosomes is known as population. A standard representation of the solution is as an array of bits. Initially many individual solutions are randomly generated to form an initial population. As the generation evolves, the population is modifies by better individuals. The selection of individuals depends on their fitness value.

Variation on individuals is determined by two operators; crossover and mutation. In crossover, a pair of parent individuals are chosen and combined to produce an offspring. Iteratively applying this procedure results in getting better solutions in the population. Finally, each solution is subject to random mutation with a small independent probability. In mutation, some random genes are replaced. This process is repeated until stopping criteria is satisfied. We can apply this genetic algorithm for not only single objective optimization but also multi-objective optimization.

Multi-objective Genetic Algorithms

Genetic Algorithms can provide very efficient solutions for multi-objective optimizations. A generic single objective GA can be modified to find a set of multiple non-

[1] blast.ncbi.nlm.nih.gov

dominated solutions in clustering method.

However, some well-known multi-objective GAs are vector evaluated GA (VEGA)[82], Niched Pareto Genetic Algorithm (NPGA)[83], Non-dominated Sorting Genetic Algorithm (NSGA)[84], Weight Based Genetic Algorithm (WBGA)[85], Strength Pareto Evolutionary Algorithm (SPEA)[86] and more. These algorithms have their own mechanism.

Design Issue of Multi-objective GA

As every other algorithms including genetic algorithms for single objective, multi-objective genetic algorithms also have some design issue. It should have a good fitness function, should preserve the diversity in solutions and a good multi-objective genetic algorithm should maintain elitism.

Fitness Function

Weighted Sum Approach The easiest way to solve multi-objective genetic algorithm is, weighted sum approach, where each objective function is multiplied with a weight. When multiple solutions are required, the same objective functions are multiplied with different weight combinations. The RWGA[87] procedure follows the weighted sum approach.

Altering objective function In this process, the population P is randomly divided into k equal sized sub-population $P_1, P_2, ..., P_k$ and each sub population is assigned a fitness value based on objective function z. Solution is selected after crossover and mutation. Algorithm VEGA [82] uses this procedure. The advantage is, it is easy to implement and computationally efficient.

Pareto Ranking Approaches Here, the individuals are ranked according to dominance rule. The population is copied to a temporary population set. From the temporary set, the most non-dominated individuals are given rank 1 and removed from the set. Then the second most non-dominated solutions are selected and given rank 2. This process continues until all the individuals are given a rank. Algorithm NSGA2[88] uses this approach to find the ranking. However, there are some other algorithms which uses a different method. The ranking is done as follows:

$$r_2(x,t) = 1 + nq(x,t)$$

, where $nq(x,t)$ is the number of solutions that are dominating solution x at generation t. In algorithm SPEA [86], an external archive E is maintained where the most non-dominated solutions found so far are stored. For each solution y belongs to E, a strength value is defined as

$$s(y,t) = \frac{np(y,t)}{Np} + 1$$

where np is the number of solutions that y dominates in P. The rank of the individual is then defined as:

$$r_3(x,t) = 1 + \sum s(y,t)$$

Accumulated ranking density strategy can also be used which is defined as:

$$r_4(x,t) = 1 + \sum r_4(y,t)$$

Diversity Measure Maintaining diversity in population is important in multi-objective GA because we need to find those solutions that are distributed over the problem domain uniformly. Following are the issues for diversity measurement.

Fitness Sharing The idea of this process is as follows; at first, the euclidean distance between each individual pairs are calculated by the given formula:

$$dz(x,y) = \sqrt{\sum_{k=1}^{K} \left(\frac{z_k(x) - z_k(y)}{z_k^{max} - z_k^{min}}\right)^2}$$

where x and y are two solution pairs, z_k^{max} and z_k^{min} are maximum and minimum value of the objective function z_k. After that, the niche count is calculated. Decision variabls can also be used to calculated the distance between two solutions.

The disadvantage of this approach is that, one has to address user-defined parameters for calculation. Algorithm MOGA[89] uses this diversity measure process.

Crowding Distance Crowding distance method can also be used as a diversity measure. Here, the individuals are ranked and then crowding distance is calculated as follows:

$$cd_k(x_{[i,k]}) = \frac{z_k(x_{[i+1,k]}) - z_k(x_{[i-1,k]})}{z_k^{max} - z_k^{min}}$$

The main advantage of the crowding approach is that population density measure does not require a user-defined parameter. In NSGA-II [88], this crowding distance measure is used as a tiebreaker in binary tournament selection technique in which two solutions x and y are randomly selected. If they are in the same non-dominated front, the

solution with a higher crowding distance is the winner. Otherwise, the solution with the lowest rank is selected. SPEA2 [90] also uses this as a diversity measure.

Cell based density In this technique, the problem domain is divided into k equal sized cells, and each individual belongs to one cell. According to the density of the cells, diversity of solutions is calculated. The algorithm PESA [91] and RDGA [92] uses this technique. The advantage of this technique is, the density measure for the whole problem domain can be represented using this technique.

Elitism In terms of single objective genetic algorithm, elitism is the best solutions found so far to survive to the next generation. In multi-objective genetic algorithm all the non-dominated solutions are elite solution. Normally there could be large number of elite solutions in multi-objective GA and it is impossible to save all the elite solutions. Most recent multi-objective GAs like [86], [93] and [94], implements elitism found far better accuracy with respect to non-elitist multi-objective GAs like in [95]. There are two approaches to achieve elitism

- maintaining elitist solutions in the population, and
- storing elitist solutions in an external archive and reintroduce them to the population

Improved Non-dominated Sorting Genetic Algorithm (NSGA-II)

As Folino *et al.* introduced in a new way for evolutionary clustering using the multi-objective genetic algorithm NSGA-II in [96] and as an improvement of [96] we are to implement an improved version of NSGA-II [88] in this paper to predict the amino acid interaction network. Deb et al. in [88] proposed this improve version of non-dominated sorting genetic algorithm in 2002. It uses non-dominated ranking approach for fitness function, crowding distance for diversity preservation and maintains elitism in population. Each time step it combines the current population and previous steps offspring and uses fast non-dominated sorting algorithm to identify the fronts. It takes the best N individuals from the best fronts to the worst, breaking the list front tie with crowding distance, where N is the population size. From this population it creates a mating pool by using binary tournament on crowding distance. N new offspring are created from this mating pool by applying crossover and mutation on the individuals.

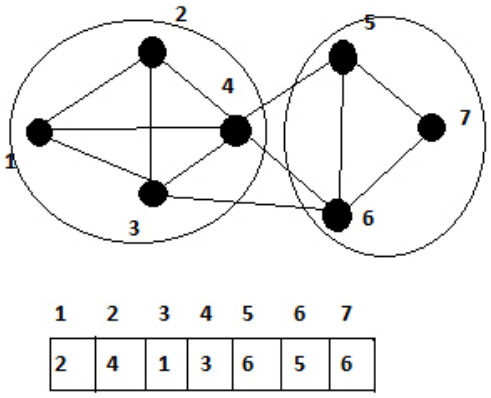


Figure 6.1: Network of 7 nodes clustered into {1,2,3,4} and {5,6,7} and their genetic representation

6.2.2 Prediction of SSE Interaction Network using Genetic Algorithm

In this paper we proposed an evolutionary clustering algorithm to predict the SSE-IN, which is a modified algorithm of the second version of strength pareto evolutionary algorithm (SPEA2) in [90]. As described before, SPEA2 preserve more better solutions than NSGA-II [88] and its diversity mechanism is better than the others, this is the reason to choose SPEA2 to implement the evolutionary clustering algorithm.

Genetic Representation

As proposed in [97], we are using a local-based adjacency representation. In this representation an individual of the population consist of N genes $g_1, \ldots, g_N$, where N is the number of nodes. Each gene can hold allele value in the range $\{1, \ldots, N\}$. Genes and alleles represents nodes in the graph $G = (V, E)$ modeling a network N. A value j assigned in $i - th$ gene interpreted as a link between node i and j and in clustering node i and j will be in the same cluster as in Figure: 6.1. In decoding step all the components are identified and nodes participating in the same component are assigned to the same cluster.

Algorithm

It takes a dynamic network $\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_T\}$, the sequence of graphs $G = \{G_1, G_2, \ldots, G_T\}$ and the number of timestamps T as input and gives a clustering

of each network $\mathcal{N}_i$ of $\mathcal{N}$ as output.

In the amino acid interaction network, total number of gene is the number SSE in the associate protein family found from the first step. Each SSE represents one gene or allele and it is the size that is the number of amino acids which compose the SSE. We represents a protein as an array of alleles. The position of an allele corresponds to the SSE position it represents in the sequence. At the same time, an incident matrix is associates for each genome.

Initialization For the first time-stamp of first input network there is no temporal relation with the previous network. The only objective function is snapshot quality or snapshot score. Thus we can apply any static clustering algorithm or trivial genetic algorithm to find the initial cluster. In this algorithm we used genetic algorithm to find the best cluster by maximizing the only objective function. As it is single objective algorithm we can find the single best cluster from this step.

Population from Graph As a first step in each time-stamp from 2nd time-stamp to T, it creates a population of random individuals. Each individual is a vector of length equal to number of nodes in the graph G^t. Genetic variant operators will be applied on this population for a fixed number of pass.

Decoding Each individual of the population and archive is decoded into component graph. As each individual gene is working as an adjacency list, if a node in x of graph is reachable from y by maintaining the edges in the individual, then x and y is in same cluster of component.

Evaluation After decoding step, each individual of the population and archive is evaluated to find the objective function values. In this algorithm the evaluation phase consist of calculate community score, which defines how better is the current clustering with respect to the given network and normalized mutual information (NMI), which defines the clustering fluctuation from the previous time-stamp. Both of the values are to be maximized.

Assign Rank Each individual of the population and archive a rank value, smaller the better. Each non-dominated individual gets the rank 0. After removing the 0 ranked individuals, give the rank 1 to the next non-dominated individuals and so on.

After giving each individuals a rank value, the individuals is sorted according to the ascending rank.

$$r(x) = \sum_{x \prec y} s(y) \tag{6.1}$$

Fitness Function There could be many individuals of in same area of solution space or objective space. If we take all these solution into account, we could loss diversity in the population. To remain the population diverse, we are using distance of $k-th$ nearest neighbor. The fitness value of each individual is the sum of its non-dominated rank and the inverse of the distance of $k-th$ nearest neighbours distance. More the distance between solutions, better the fitness functions value.

$$m(x) = \left(\sigma_x^k + 1\right)^{-1} \tag{6.2}$$

where σ_x^k is the distance between individual x and its $k-th$ nearest neighbour. To calculate the distance between chromosome, we have to take into account the three objectives, atomic distance of amino acids, torsion angles and hydrophobicity.

Population Selection After evaluating fitness values of each of the population and archive, the best individuals are selected as a new population. From the total individuals of population and archive population size individuals are selected as new population. From the rank 0 to the highest rank, all the individuals are added if number of population of this rank is not exceeding the current population size. If it is exceeding, then some individuals are truncated according to the value of each individuals.

Mating Pool Creation After selecting the new population, a mating pool is created of pool size from the new population to apply the genetic variation operators. To choose the mating pool, binary tournament with replacement has been used in this algorithm. According to binary tournament, two individuals are randomly selected from the new population and the better fitness valued individual is chosen for the mating pool.

Genetic Variation Operators Genetic operators are used to create offspring from parent or mating pool. As other genetic algorithms, in this algorithm two widely used genetic variation operators have been used. These are crossover and mutation.

Crossover Crossover is the operator which is used to create offspring from two parents. The offspring bear the genes of each parent. As a genetic variation operator there is very high probability to crossover occurs other than mutation. In this algorithm

Parent1:	4	3	2	2	6	5	6
Parent2:	3	3	1	5	4	7	6
Mask :	0	1	1	0	0	1	1
Offspring:	4	3	1	2	6	7	6

Table 6.1: Example of Uniform Crossover

we are using uniform crossover. A random bit vector of length of number of the node in the current graph is created. If $i-th$ bit is 0 then the value of the $i-th$ gene comes from the first parent otherwise it comes from the $i-th$ gene of second parent. As each of the parents holding true adjacency information, the offspring will also hold it.

Mutation One of the most widely used variation operator in genetic algorithm, which perform the operation in a single individual is mutation. Though the probability of mutation is normally very low, but it is the best way to make small variation in the individual. To mutate and create a offspring, some position of the of the individuals are chosen randomly and changed to other values. But the value should be one of its neighbour in the current graph.

Topological operator A topological operator is used to exclude incompatible population generated by the algorithm. We compute the diameter, the characteristic path length and the mean degree to evaluate the average topological properties of the family for the particular SSE number.

Archive There is an archive of individual which saves the best solutions as a property of elitism, which is the main differential property of SPEA2 algorithm. After variation operators over the offspring becomes the new population and the old populations are saved to the archive. To fit the archive size, the truncation mechanism used here also. By saving archive, this algorithm maintains the elitism.

Algorithm 1 Multi-objective genetic algorithm to predict SSE interaction network

1: **Input:** A protein sequence, T = total time steps, N_E = Archive size, N_P = Population Size
2: **Output:** A predicted incident matrix M and clustering for each network $\mathcal{N}^i$ of $\mathcal{N}$

3: Use BLAST to find a associate protein family of the given sequence from PDB.
4: Generate initial cluster $\mathcal{CR}_1 = \{C_1^1, \ldots, C_k^1\}$ of the network $\mathcal{N}^1$ with number of vertex equal to number of SSE of the associate protein family
5: **for** $t = 2$ to T **do**
6: Create initial population of random individual P_0 and set $E_0 = \emptyset, i = 0$
7: **loop**
8: Decode each individual of $P_i \cup E_i$
9: Evaluate each individual of $P_i \cup E_i$ to find rank and density value using Equation 6.1 and 6.2
10: Assign fitness value to each individual, as the sum of rank and inverse of density value
11: Copy all non-dominated solutions to E_{i+1}
12: **if** $|E_{i+1}| > N_E$ **then**
13: truncate $|E_{i+1}| - N_E$ solutions according to topological property
14: **else**
15: copy best $N_E - |E_{i+1}|$ dominated solutions according to their fitness value and topological property
16: **end if**
17: **if** If stopping criteria satisfied **then**
18: **return** non-dominated solutions in E_{i+1}
19: **else**
20: Select some individuals from— E_{i+1} for mating pool as parents using binary tournament with replacement
21: Apply crossover and mutation operators to the mating pool to create N_P offspring solutions and copy to P_{i+1}
22: $i := i + 1$
23: **end if**
24: **end loop**
25: From the returned solution in E take the best cluster according to the highest modularity value
26: **end for**

6.3 Ant Colony Optimization (ACO) to Predict Interactions

After predicting the SSE-IN network we have to identify the interactions involve between the amino acids in the folded protein. We have used an ant colony optimization (ACO) approach to select and predict the edges which link different SSE's, considering about the correction of the matrix of motifs previously predicted.

We have build a two steps algorithm as the hierarchical structure of the SSE-IN.

- In interaction, consider each pair of SSE's separately. This is the local step. We use an ant colony algorithm to identify the suitable interactions between amino acids belonging to these SSE's.
- A global ant colony algorithm is run to predict the interaction between amino acids from different SSE-IN.

6.3.1 Parameters for Interaction Network Prediction

To predict the interactions, firstly we have to know how many edges to be add in the network and which nodes we should consider in interactions. To find and evaluate these parameters, the template proteins from the associate family is incorporated.

Some template proteins is selected from the associate family whose SSE number is same as the sequence to predict the edge rate of the sequence and represent them as chromosome or array of alleles as in the multi-objective genetic algorithm. Thus, we build an comparative model to compute the edge ratio, which is used to fold the sequence SSE-IN.

The average chromosome is calculated from all the template proteins in associate protein family. The distance between two chromosome is used as discussed in the previous section to compare the sequence with the average chromosome. We add up the distance allele by allele to obtain a distance between the sequence and the average family chromosome. After that, the cumulated size is calculated by adding up the the chromosome cell values. If the distance is less than 20% of the sequence cumulated size and the average family chromosome then the sequence is closer to the template protein. Then we compute the average edge rate in the closer protein to add the initial edges in the disconnected network of the sequence. If a sequence can not be found closer to the template one we add the sequence with the average family chromosome and starts again the same procedure.

We do the same procedure to find the designation of the vertices, which vertices should interact with each other as they also use comparative model.

To define, which edges link two SSE's, we consider the following problem. Let $X = x_1, x_2, \ldots x_n$ and $Y = y_1, y_2, \ldots y_m$ be two SSEs in interaction. We want to add e edges among the $n \times m$ possible combinations. For $i \in [1, n]$ and $j \in [1, m]$ the probability to interact the amino acid x_i with y_j, is correlated with the occurrence matrix of the predicted edges ratios, represented by $Q(x_i, y_j)$ and we can assume $s_{ij} \sim Q(x_i, y_j)$. To add approximately e edges, we need

$$\sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} = e \tag{6.3}$$

and

$$s_{ij} = \frac{eQ(x_i, y_j)}{\sum_{p=1}^{n} \sum_{q=1}^{m} Q(x_p, y_q)} \tag{6.4}$$

6.3.2 Algorithm

The prediction of interaction network consists of two approach, local and global algorithm.

Local Algorithm

The local algorithm is used to predict the suitable short cut edges between pair of SSEs in the network. Thus, we differentiate each pair of SSEs which have connection and build a graph where each vertex of the first SSE is connected to each vertex of the other SSE. The connection or the edges are weighted and the weight is s_{ij}, discussed before in this section. Then we used an ant colony approach consists of an ant number equals to the number of vertices in two SSE. The ant system has to reinforce the suitable edges between the SSEs. We use these edges in the global algorithm which describes in the next section.

The local ant colony algorithm first creates n ants which is total number of vertices in the two SSEs related in the search. For an ant to be positioned we choose a random vertex of the to SSE involved and place it. All the n ants are positioned this way and two ants can share same vertex. An ant in vertex i will choose the vertex j with probability p_{ij}, defined as follows:

$$p_{ij} = \frac{{\tau_{ij}}^{\alpha} \cdot {s_{ij}}^{\beta}}{\sum_{k \in V(i)} {\tau_{ij}}^{\alpha} \cdot {s_{ij}}^{\beta}} \tag{6.5}$$

The weight s_{ij} also called heuristic vector, calculated before. If the vertices i and j are in the same SSE, then the edge between these two vertices has weight equal to the average weight of the shortcut edges:

$$\overline{w} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} \tag{6.6}$$

After each move of an ant we update the pheromone value on the inter-SSE edges using the formula,

$$\tau_{ij} = (1 - \rho)\tau_{ij} + n_{ij}\Delta\tau \tag{6.7}$$

where n_{ij} is the number of ants that moved on the edge (i, j) and $\Delta\tau$ is the quantity of pheromone dropped by each ant. As far as the edges belonging to the same SSE are concerned, we keep the pheromone rate on them equals to the average pheromone rate on the inter-SSE edges

$$\overline{\tau} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \tau_{ij} \tag{6.8}$$

Ants move inside an SSE randomly, described above, on the other hand if they decide to change the SSE they are guided by the edge weight and the weight is guided by the pheromone value. The algorithms stops after a predefined number of iteration or the maximum pheromone rate is e time bigger than the average pheromone rate on the edge. After the execution of the algorithm we keep the edges whose pheromone quantity exceeds a threshold λ_{min}.

Algorithm 2 Local algorithm to find Inter-SSE edges

```
1: Input: The predicted network from the multiobjective genetic algorithm
2: Output: Predicted inter-SSE edges
3: Create n ants, where n is the total number of nodes in process
4: while stopping criteria does not meet do
5:    for all ant a do
6:       moveAnt(a)
7:    end for
8:    updatePheromone()
9: end while
10: selectEdges(λ_min)
```

Global Algorithm

After the local algorithm execution, we get the SSE-IN composed of these specific inter-SSE edges. The global algorithm will keep the number of edges exactly E_p, which was predicted before. As the local one, the global algorithm uses the ant colony approach with the number of vertices equal to the SSE-IN vertex number. The ants movements contribute to emerge the specific short cuts whose only a number E_p is kept. We rank the short cut edges as a function of the pheromone quantity to extract the E_p final short cuts. Finally, we measure the resulting SSE-INs by topological metrics to accept it or not.

Algorithm 3 Global algorithm to predict edges into SSEs

```
1: Input: The network with predicted edges E_s from local algorithm and E_p, number
   edges to predict
2: Output: The network with total E_p edges
3: buildSSEIN(E_s)
4: create n ants
5: while stopping criteria does not meet do
6:    for all ant a do
7:       moveAnt(a)
8:    end for
9:    updatePheromone()
10: end while
11: selectEdges(E_p)
```

We compute the diameter, the characteristic path length and the mean degree to evaluate the average topological properties of the family for a particular SSE number. Then, after we have built the sequence SSE-IN, we compare its topological properties with the template ones. We admit an error up to 20% to accept the built sequence SSE-IN. If the built SSE-IN is not compatible, it is rejected. We compare the predicted value, denoted E_p , with the real value, denoted E_R,

$$AC = 1 - \frac{|E_R - E_p|}{E_p} \tag{6.9}$$

where AC is the accuracy of the prediction.

Chapter 7

Performance Analysis

As we have described the proposed algorithm in the chapter 6 and the algorithm is robust and covering more criteria comparing to other state of the art algorithms, the algorithm should also perform better than other algorithms. In this chapter we will discuss about result and performance comparison with this proposed algorithm and some previous algorithms. Before the performance analysis performance metrics will be discussed briefly.

7.1 Analysis of Genetic Algorithm as Multi-objective Optimization

In order to test the performance of proposed multi-objective genetic algorithm, we randomly pick three chromosomes from the final population and we compare their associated matrices to the sequence SSE-IN adjacency matrix. To evaluate the difference between two matrices, we use an error rate defined as the number of wrong elements divided by the size of the matrix. The dataset we use is composed of 698 proteins belonging to the All alpha class and 413 proteins belonging to the All beta class. A structural family has been associated to this dataset as in [80].

All alpha class has an average error rate of 14.6% and for the *All beta* class it is 13.1% and the maximum error rate shown in the experiment is 22.9%. Though, the error rate depends on other criteria like the three objectives described before but according to the result we can firmly assert that the error rate depends on the number of initial population, more the number of initial population less the error rate. With sufficient number of individuals in the initial population we can ensure the genetic diversity as well as the improved SSE-IN prediction. When the number of initial population is at

least 15, the error rate is always less than 10%.

As compared to the work in [2], we can claim better and improved error rate in this part of SSE-IN prediction algorithm.

7.2 Analysis of Ant Colony Optimization

We have experimented and tested this part of our proposed method according to the associated family protein because the probability of adding edge is determined by the family occurrence matrix. We have used the same dataset of sequences whose family has been deduced.

For each protein, we have done 150 simulations and when the topological properties are became compatible to the template properties of the protein we accepted the built SSE-IN. The results are shown in Tab. 7.1. The score is the percentage of correctly predicted short cut edges between the sequence SSE-IN and the SSE-IN we have reconstructed [2]. In most cases, the number of edges to add were accurate according to the plot 7.1. From this we can percept that, global interaction scores depends on the local algorithm lead for each pair of SSEs in contact. The plot, in Figure 7.2, confirms this tendency, if the local algorithm select at least 80% of the correct short cut edges, the global intersection score stays better than the 80% and evolves around 85% for the All alpha class and 73% for the All beta class.

After the discussion we can say that, though for the big protein of size more than 200 amino acids the average score decreases, but in an average the score remains for the global algorithm around 80%.

7.3 Algorithm Complexity

Our proposed algorithm is independent of specific time bound. Both the optimization algorithm used as multi-objective genetic algorithm and ant colony algorithm, is iteration based. We can stop the algorithm at any time. Though the result of the algorithm depends on the number of iteration but if we give sufficient amount of iteration it provides good result. In compare to other state of art algorithms, those uses exponential complexity algorithm, our is linear in terms of time and memory.

Class	SCOP Family	Number of Proteins	Protein Size	Score	Average Deviation
All Alpha	46688	17	27 - 46	83.973	3.277
	47472	10	98 - 125	73.587	12.635
	46457	25	129 - 135	76.125	7.489
	48112	11	194 - 200	69.234	14.008
	48507	18	203 - 214	66.826	5.504
	46457	16	241 - 281	63.281	17.025
	48507	20	387 - 422	62.072	9.304
All Beta	50629	6	54 - 66	79.635	2.892
	50813	11	90 - 111	74.006	4.428
	48725	24	120 - 124	80.881	7.775
	50629	13	124 - 128	76.379	9.361
	50875	14	133 - 224	77.959	10.67

Table 7.1: Folding a SSE-IN by an ant colony approach. The score measures the interaction between the effective short cut edges and the predicted one. For small proteins the scores are better than 75% because the number of SSE is weak and the global algorithm is less dependent on local one. The algorithm parameter values are: $\alpha = 25, \beta = 12, \rho = 0.7, \Delta\tau = 4000, e = 2, \lambda_{min} = 0.8$.

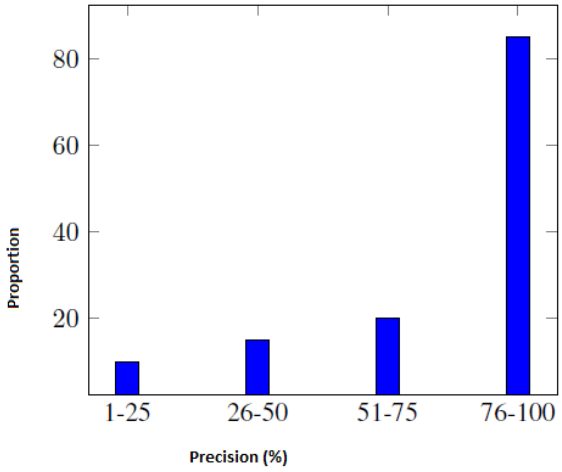


Figure 7.1: Precision of number of edges to be added in All alpha class

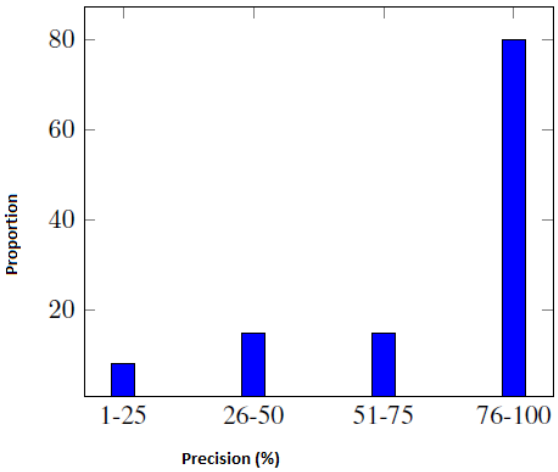


Figure 7.2: Precision of number of edges to be added in All beta class

Chapter 8

Conclusion

This chapter summarizes the work presented in this thesis and figure out the advantages and limitations of the proposed algorithm. It also discussed about the scope of future works related to the proposed algorithm.

8.1 Discussion

We have proposed an computational solution to an biological problem. We have described how we can formulate a biological problem like folding protein into optimization and graph theory problem. The formulation consist of finding the interactions between secondary structure element (SSE) network and interaction between amino acids of the protein. The first problem was solve by an multi-objective genetic algorithm and the second one solve by ant colony optimization approach.

As discussed in the chapter 7, we can claim that our proposed algorithm gives more accurate result in terms of accuracy and score to predict the amino acid interaction network.

8.2 Future Works

In this section a brief discussion on the improvement of this algorithm and probable future works.

1. Improvement in Genetic Representation: As a genetic representation of the individuals of a network, we have used vector. Instead a matrix or adjacency list can be used. Obviously it will increase the complexity of the algorithm in terms of time and memory.

2. Improvement in Data Structure: To preserve the diversity function we need k-th nearest neighbour of each individual. To find this we did not use any good data structure. We can use kd-tree to improve this.

3. Improvement in functions: The objective function as well as other function like diversity measurement can be improved much.

4. Parallelize: The proposed algorithm in this paper can be parallelized. Then it can be a fast and more accurate algorithm. The big limitation of this algorithm, run on large network, could be reduced by parallelizing this algorithm.

Our next step would be, come up with better genetic representation, use of appropriate and advanced data structure and a powerful parallel algorithm.

References

[1] E. Pennisi. A low number wins the genesweep pool. In *Science*, volume 300, page 1484, 2003.

[2] O. Gaci. How to fold amino acid interaction networks by computational intelligence methods. In *BioInformatics and BioEngineering (BIBE), 2010 IEEE International Conference on*, pages 150 – 155, 2010.

[3] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, *KDD*, pages 554–560. ACM, 2006.

[4] M.N. Murty A. K. Jain and P.J. Flynn. Data Clustering . In *ACM Comuting Surveys*, volume 31, pages 264–296, 1999.

[5] L. Li N. V. Dokholyan, F. Ding, and E. I. Shakhnovich. Topological determinants of protein folding. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 99, pages 8637–8641, 2002.

[6] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, TN Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[7] Ingrid Wagner and Hans Musso. New naturally occurring amino acids. *Angewandte Chemie International Edition in English*, 22(11):816–828, 1983.

[8] Debnath Pal and Pinak Chakrabarti. On residues in the disallowed region of the ramachandran map. *Biopolymers*, 63(3):195–206, 2002.

[9] Carl Branden, John Tooze, et al. *Introduction to protein structure*, volume 2. Garland New York, 1991.

[10] John Moult. A decade of casp: progress, bottlenecks and prognosis in protein structure prediction. In *Current Opinion in Structural Biology*, volume 15, pages 285–289, 2005.

[11] John Moult, Tim Hubbard, Krzysztof Fidelis, and Jan T. Pedersen. Critical assessment of methods of protein structure prediction (casp)–round ix. In *Proteins-structure Function and Bioinformatics - PROTEINS*, volume 79, pages 1–5, 2011.

[12] Alexey G. Murzin, Steven E. Brenner, Tim Hubbard, and Cyrus Chothia. Scop: A structural classification of proteins database for the investigation of sequences and structures. In *Journal of Molecular Biology - J MOL BIOL*, volume 247, pages 536–540, 1995.

[13] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb C. Miller, and David J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. In *Nucleic Acids Research - NAR*, volume 25, pages 3389–3402, 1997.

[14] David T. Jones. Predicting novel protein folds by using fragfold. In *Proteins-structure Function and Bioinformatics - PROTEIN*, volume 45, pages 127–132, 2001.

[15] David T. Jones and Liam J. McGuffin. Assembling novel protein folds from super-secondary structural fragments. In *Proteins-structure Function and Bioinformatics - PROTEINS*, volume 53, pages 480–485, 2003.

[16] Kim T. Simons, Rich Bonneau, Ingo Ruczinski, and David Baker. Ab initio protein structure prediction of casp iii targets using rosetta. In *Proteins-structure Function and Bioinformatics - PROTEINS*, volume 37, pages 171–176, 1999.

[17] Richard Bonneau, Jerry Tsai, Ingo Ruczinski, Dylan Chivian, Carol Rohl, Charlie E. M. Strauss, and David Baker. Rosetta in casp4: Progress in ab initio protein structure prediction. In *Proteins-structure Function and Bioinformatics - PROTEINS*, volume 45, pages 119–126, 2001.

[18] Adam Zemla. Lga: A method for finding 3d similarities in protein structures. In *Nucleic Acids Research - NAR*, volume 31, pages 3370–3374, 2003.

[19] Kryshtafovych A, Fidelis, and John Moult. Casp9 results compared to those of previous casp experiments. In *Proteins-structure Function and Bioinformatics - PROTEINS*, volume 79, pages 196–207, 2011.

[20] Valerio Mariani, Florian Kiefer, Tobias Schmidt, Juergen Haas, and Torsten Schwede. Assessment of template based protein structure predictions in casp9. In *Proteins-structure Function and Bioinformatics - PROTEINS*, volume 79, pages 37–58, 2011.

[21] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S. Marks, Chris Sander, Riccardo Zecchina, Jose' N. Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. In *National Academy of Sciences of the USA*, volume 108, pages E1293–E1301, 2011.

[22] Debora S. Marks, Lucy J. Colwell, Robert Sheridan, Thomas A. Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3d structure computed from evolutionary sequence variation. In *PLOS One*, volume 6, page E28766, 2011.

[23] D.T. Jones, D.W. Buchan, D. Cozzetto, and M. Pontil. Psicov: Precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. In *Bioinformatics*, volume 28, pages 184–190, 2012.

[24] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. In *Nature Methods - NAT METHODS*, volume 9, pages 173–175, 2011.

[25] Johannes Söding, Andreas Biegert, and Andrei N. Lupas. The hhpred interactive server for protein homology detection and structure prediction. In *Nucleic Acids Research - NAR*, volume 33, pages 244–248, 2005.

[26] Philip Bradley, Kira M. S. Misura, and David Baker. Toward high-resolution de novo structure prediction for small proteins. In *Science*, volume 309, pages 1868–1871, 2005.

[27] Adam Zemla. Lga: a method for finding 3d similarities in protein structures. In *Nucleic Acids Research - NAR*, volume 31, pages 3370–3374, 2003.

[28] Ken A. Dill, S. Banu Ozkan, M. Scott Shell, and Thomas R. Weikl. The protein folding problem. In *Annual Review of Biophysics*, volume 37, pages 289–316, 2008.

[29] Frances Allen, G Almasi, Wanda Andreoni, D Beece, Bruce J. Berne, A Bright, Jose Brunheroto, Calin Cascaval, J Castanos, Paul Coteus, et al. Blue gene: a vision for protein science using a petaflop supercomputer. *IBM Systems Journal*, 40(2):310–327, 2001.

[30] Michael Shirts, Vijay S Pande, et al. Screen savers of the world unite. *COMPUTING*, 10:43, 2006.

[31] Christopher D Snow, Houbi Nguyen, Vijay S Pande, and Martin Gruebele. Absolute comparison of simulated and experimental protein-folding dynamics. *Nature*, 420(6911):102–106, 2002.

[32] David E Shaw, Martin M Deneroff, Ron O Dror, Jeffrey S Kuskin, Richard H Larson, John K Salmon, Cliff Young, Brannon Batson, Kevin J Bowers, Jack C Chao, et al. Anton, a special-purpose machine for molecular dynamics simulation. In *ACM SIGARCH Computer Architecture News*, volume 35, pages 1–12. ACM, 2007.

[33] J Andrew McCamm0n. Dynamics of folded proteins. *Nature*, 267:16, 1977.

[34] David E Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, Michael P Eastwood, Joseph A Bank, John M Jumper, John K Salmon, Yibing Shan, et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

[35] Rong Chen, Julian Mintseris, Joël Janin, and Zhiping Weng. A protein–protein docking benchmark. *Proteins: Structure, Function, and Bioinformatics*, 52(1):88–91, 2003.

[36] J Peter Guthrie. A blind challenge for computational solvation free energies: introduction and overview. *The Journal of Physical Chemistry B*, 113(14):4501–4507, 2009.

[37] Mayako Michino, Enrique Abola, GPCR Dock, et al. Community-wide assessment of gpcr structure modelling and ligand docking: Gpcr dock 2008. *Nature Reviews Drug Discovery*, 8(6):455–463, 2009.

[38] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[39] Sarel J Fleishman, Timothy A Whitehead, Damian C Ekiert, Cyrille Dreyfus, Jacob E Corn, Eva-Maria Strauch, Ian A Wilson, and David Baker. Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science*, 332(6031):816–821, 2011.

[40] Brian Kuhlman, Gautam Dantas, Gregory C Ireton, Gabriele Varani, Barry L Stoddard, and David Baker. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302(5649):1364–1368, 2003.

[41] Justin B Siegel, Alexandre Zanghellini, Helena M Lovick, Gert Kiss, Abigail R Lambert, Jennifer L St Clair, Jasmine L Gallaher, Donald Hilvert, Michael H Gelb, Barry L Stoddard, et al. Computational design of an enzyme catalyst for a stereoselective bimolecular diels-alder reaction. *Science*, 329(5989):309–313, 2010.

[42] Mihai L Azoitei, Bruno E Correia, Yih-En Andrew Ban, Chris Carrico, Oleksandr Kalyuzhniy, Lei Chen, Alexandria Schroeter, Po-Ssu Huang, Jason S McLellan, Peter D Kwong, et al. Computation-guided backbone grafting of a discontinuous motif onto a protein scaffold. *Science*, 334(6054):373–376, 2011.

[43] Samuel H Gellman. Foldamers: a manifesto. *Accounts of Chemical Research*, 31(4):173–180, 1998.

[44] W Seth Horne and Samuel H Gellman. Foldamers with heterogeneous backbones. *Accounts of chemical research*, 41(10):1399–1408, 2008.

[45] Kent Kirshenbaum, Ronald N Zuckermann, and Ken A Dill. Designing polymers that mimic biomolecules. *Current opinion in structural biology*, 9(4):530–535, 1999.

[46] Byoung-Chul Lee, Ronald N Zuckermann, and Ken A Dill. Folding a nonbiological polymer into a compact multihelical structure. *Journal of the American Chemical Society*, 127(31):10999–11009, 2005.

[47] Sarah A Fowler and Helen E Blackwell. Structure–function relationships in peptoids: recent advances toward deciphering the structural requirements for biological function. *Organic & biomolecular chemistry*, 7(8):1508–1524, 2009.

[48] Byoung-Chul Lee, Tammy K Chu, Ken A Dill, and Ronald N Zuckermann. Biomimetic nanostructures: Creating a high-affinity zinc-binding site in a folded

nonbiological polymer. *Journal of the American Chemical Society*, 130(27):8847–8855, 2008.

[49] Neel H Shah and Kent Kirshenbaum. Photoresponsive peptoid oligomers bearing azobenzene side chains. *Org. Biomol. Chem.*, 6(14):2516–2521, 2008.

[50] Ki Tae Nam, Sarah A Shelby, Philip H Choi, Amanda B Marciel, Ritchie Chen, Li Tan, Tammy K Chu, Ryan A Mesch, Byoung-Chul Lee, Michael D Connolly, et al. Free-floating ultrathin two-dimensional crystals from sequence-specific peptoid polymers. *Nature materials*, 9(5):454–460, 2010.

[51] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1):309–320, 2000.

[52] Stanley Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[53] Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltan N Oltvai, and A-L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[54] Ray Solomonoff and Anatol Rapoport. Connectivity of random nets. *The bulletin of mathematical biophysics*, 13(2):107–117, 1951.

[55] P ERDdS and A R&WI. On random graphs i. *Publ. Math. Debrecen*, 6:290–297, 1959.

[56] Paul Erd6s and A Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.

[57] Réka Albert and Albert-László Barabási. Topology of evolving networks: local events and universality. *Physical review letters*, 85(24):5234, 2000.

[58] Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[59] Duncan J Watts. *Small worlds: the dynamics of networks between order and randomness*. Princeton university press, 1999.

[60] Mark EJ Newman. The structure and function of networks. *Computer Physics Communications*, 147(1):40–45, 2002.

[61] Luis A Nunes Amaral, Antonio Scala, Marc Barthélémy, and H Eugene Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences*, 97(21):11149–11152, 2000.

[62] K-I Goh, B Kahng, and D Kim. Universal behavior of load distribution in scale-free networks. *Physical Review Letters*, 87(27):278701, 2001.

[63] R. Diestel. Graph Theory: Electronic Edition 2000. 2000.

[64] Thomas F Coleman and Jorge J Moré. Estimation of sparse jacobian matrices and graph coloring blems. *SIAM journal on Numerical Analysis*, 20(1):187–209, 1983.

[65] Omar Gaci, Stefan Balev, et al. Proteins: From structural classification to amino acid interaction networks. *Proceedings of the BIOCOMP 08*, 2(1):728–734, 2008.

[66] Amit Ghosh, KV Brinda, and Saraswathi Vishveshwara. Dynamics of lysozyme structure network: probing the process of unfolding. *Biophysical journal*, 92(7):2523–2535, 2007.

[67] Omar Gaci and Stefan Balev. The small-world model for amino acid interaction networks. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*, pages 902–907. IEEE, 2009.

[68] Usha K Muppirala and Zhijun Li. A simple approach for protein structure discrimination based on the network pattern of conserved hydrophobic residues. *Protein Engineering Design and Selection*, 19(6):265–275, 2006.

[69] KV Brinda and Saraswathi Vishveshwara. A network representation of protein structures: implications for protein stability. *Biophysical journal*, 89(6):4159–4170, 2005.

[70] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*, volume 4. Oxford university press New York, 1999.

[71] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. Inspiration for optimization from social insect behaviour. *Nature*, 406(6791):39–42, 2000.

[72] Julia Handl, Joshua Knowles, and Marco Dorigo. Ant-based clustering and topographic mapping. *Artificial Life*, 12(1):35–62, 2006.

[73] Mike Campos, Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Dynamic scheduling and division of labor in social insects. *Adaptive Behavior*, 8(2):83–95, 2000.

[74] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010.

[75] Marco Dorigo. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*, 1992.

[76] Marco Dorigo, Vittorio Maniezzo, Alberto Colorni, and Vittorio Maniezzo. Positive feedback as a search strategy. 1991.

[77] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29–41, 1996.

[78] J-L Deneubourg, Serge Aron, Simon Goss, and Jacques Marie Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of insect behavior*, 3(2):159–168, 1990.

[79] Christian Blum. Ant colony optimization: Introduction and recent trends. *Physics of Life reviews*, 2(4):353–373, 2005.

[80] Omar Gaci. Building a topological inference exploiting qualitative criteria. *Evolutionary Bioinformatics*, 2010.

[81] Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, 1989.

[82] Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. In *Proc. of the international conference on genetic algorithm and their applications*, 1985.

[83] Nafpliotis N Horn J and Goldberg DE. A niched pareto genetic algorithm for multiobjective optimization. In *Proc. of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence*, Orlando, FL, USA, June 1994.

[84] Srinivas N. and Deb K. Multiobjective optimization using non-dominated sorting in genetic algorithms. 1994.

[85] Hajela P and lin C-y. Genetic search strategies in multicriterion optimal design. pages 99–107, 1992.

[86] Zitzler E and Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. pages 257–271, 1999.

[87] Murata T and Ishibuchi H. Moga: multi-objective genetic algorithms. In *Proc. of the 1995 IEEE international conference on evolutionary computation*, Perth,Australia, 1995.

[88] Deb K, Agrawal S, Pratap A, and Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Proc. of 6th international conference on parallel problem solving on nature*, Paris,France, September 2000. Springer.

[89] Fonseca CM and Fleming PJ. Multiobjective genetic algorithms. In *IEE colloquium on 'Genetic Algorithms for Control Systems Engineering'*, London,UK, May 1993.

[90] Marco Laumanns Eckart Zitzler and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Zurich, Switzerland, 2001.

[91] Corne DW, Knowles JD, and Oates MJ. The pareto envelope-based selection algorithm for multiobjective optimization. In *Proc. of 6th international conferennce on parallel problem solving from Nature*, Paris,France, September 2000. Springer.

[92] Yen GG and Lu H. Rank-density-based multiobjective genetic algorithm and benchmark test function study. pages 325–343, 2003.

[93] Deb K. Multi-objective optimization using evolutionary algorithms. NewYork,USA, 2001. Willey.

[94] Van Veldhuizen DA and Lamont GB. Multiobjective evolutionary algorithms: analyzing the state-of-art. pages 125–147, 2000.

[95] Jensen MT. Reducing the run-time complexity of multiobjective eas: the nsga-ii and other algorithms. pages 503–515, 2003.

[96] Francesco Folino and Clara Pizzuti. A multiobjective and evolutionary clustering method for dynamic networks. Institute for High Performance Computing and Networking (ICAR) and Italian National Research Council, 2010.

[97] Y.J. Park and M.S. Song. A genetic algorithm for clustering problems. In *Proc. of 3rd Annual Conference on Genetic Algorithm*, pages 2–9, 1989.