COMPUTATIONAL APPROACHES TO UNDERSTANDING THE PROTEIN STRUCTURE

by
PELİN AKAN

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

Sabanci University
July 2002

ABSTRACT

This thesis is composed of two different parts, aiming to predict and understand the protein structure from their contact maps. In the first part, residue contacts of a protein are predicted using neural networks in order to obtain structural constraints for the three-dimensional structure. Physical and chemical properties of residues and their primary sequence neighbors are used for the prediction. Our predictor can predict 11% of the contacting residuees with a false positive ratio of 2% and it performs 7 times better than a random predictor.

In the second part, a new method is developed to model a protein as a network of its interacting residues. Small-world network concept is utilized to interpret the parameters of residue networks. It is concluded that proteins are neither regular nor randomly packed but between these two extremes. Such a structure gives the proteins the ability for fast information relay between their residues. They can undergo necessary conformational changes for their functions on very short time scales. Also, residuee networks are shown to obey a truncated power-law degree distribution instead of being scale-free. This shows that proteins have fewer structurally weak points, whose failure would be total damage for the system. This finding conforms to evolutionary plasticity of proteins: Having a low number of weak points makes the mild DNA mutations to be translated into the protein structure as highly tolerable.

ÖZET

Bu tez çalışmasında, proteinlerin temas matrisleri kullanılarak yapıları tahmin edilmeye ve anlaşılmaya çalışılmıştır. İki bölümden oluşan bu tezin ilk bölümünde, sinir ağları kullanılarak, proteinler için yapısal sınırlamalar bulmak amacıyla residü temasları tahmin edilmiştir. Bu tahminler için residülerin fiziksel ve kimyasal özellikleri, ve birincil sekanstaki komşuları kullanılmıştır. Sonuç olarak, birbiriyle temas eden residülerin % 11'i doğru, temas etmeyen residüların % 2'si yanlış tahmin edilmiştir, ve rastlantısal bir tahminden 7 kat daha iyi sonuçlar elde edilmiştir.

İkinci bölümde, bir proteini, temas eden residülarından oluşan bir ağ olarak modellemek için yeni bir yöntem geliştirilmiştir. Bu ağların yapısal özelliklerini anlayabilmek için küçük-dünyalar fikri kullanılmıştır. Gösterilmektedir ki, residüler proteinler içinde ne düzgün ne de rastlantısal bir şekilde organizedir, küçük-dünya ağlarına benzer bir organizasyona sahiptirler. Böyle bir yapı, proteinleri çok kısa zamanlar dahilinde büyük yapısal değişimler geçirebilmesini olanaklı kılmaktadır. Ayrıca, residü ağlarının komşu sayısı dağılımları da kesik ölçeksiz dağılımlar şeklindedir. Bu da proteinlerin çok az sayıda yapısal hassas noktalar içerdiğini göstermektedir. Proteinlerin evrim sürecinde sayısız biyolojik işlevi gerçekleştirebilecek şekildeki değişimleri bu sonucu desteklemektedir. Bunun nedeni,, az sayıda hassas noktanın varlığı küçük DNA mutasyonlarının proteinlerinin yapısına yansımasına olanak sağlamasıdır.

ACKNOWLEDGEMENTS

I owe my deepest and sincere thanks to Assoc. Prof. Canan Baysal. I consider myself lucky to have met such an intelligent and dynamic person at the right time in my education. She contributed a great deal of time and energy to this thesis and has become a constant source of advice, support and love.

Special thanks goes to my dearest, Murat Kaymaz, whose love and friendship have been essential for my success.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

C_α	Central carbon atom attached to a hydrogen, an amino group, a carboxyl group and the side chain group in an amino acid
C_β	Side chain carbon atom bonded to C_α atom of a residue
Å	Angstrom

LIST OF ABBREVIATIONS

<A>	Average accuracy of a neural network
All pr.	All proteins in the dataset
C	Clustering density
C. elegans	Caenorhabditis elegans
CC	Correctly predicted contacting residues by the neural network
COF	A protein dataset comprising 225 proteins
COLD	Constrained optimization with limited deviations
DT	Delaunay Triangulation
FP	Non-contacting residues predicted as contacts by the neural network
HOT	Highly optimized tolerance
L	Characteristic path length
LRN	A protein dataset comprising 196 proteins
N1	Neural network 1 architecture
N2	Neural network 2 architecture
N3	Neural network 3 architecture
N4	Neural network 4 architecture
NN	Artificial neural network
R	Improvement of the prediction over a random predictor
TS97	A protein dataset comprising 176 proteins
WWW	World Wide Web

COMPUTATIONAL APPROACHES TO UNDERSTANDING THE PROTEIN STRUCTURE

by
PELİN AKAN

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

Sabanci University
July 2002

ABSTRACT

This thesis is composed of two different parts, aiming to predict and understand the protein structure from their contact maps. In the first part, residue contacts of a protein are predicted using neural networks in order to obtain structural constraints for the three-dimensional structure. Physical and chemical properties of residues and their primary sequence neighbors are used for the prediction. Our predictor can predict 11% of the contacting residuees with a false positive ratio of 2% and it performs 7 times better than a random predictor.

In the second part, a new method is developed to model a protein as a network of its interacting residues. Small-world network concept is utilized to interpret the parameters of residue networks. It is concluded that proteins are neither regular nor randomly packed but between these two extremes. Such a structure gives the proteins the ability for fast information relay between their residues. They can undergo necessary conformational changes for their functions on very short time scales. Also, residuee networks are shown to obey a truncated power-law degree distribution instead of being scale-free. This shows that proteins have fewer structurally weak points, whose failure would be total damage for the system. This finding conforms to evolutionary plasticity of proteins: Having a low number of weak points makes the mild DNA mutations to be translated into the protein structure as highly tolerable.

ÖZET

Bu tez çalışmasında, proteinlerin temas matrisleri kullanılarak yapıları tahmin edilmeye ve anlaşılmaya çalışılmıştır. İki bölümden oluşan bu tezin ilk bölümünde, sinir ağları kullanılarak, proteinler için yapısal sınırlamalar bulmak amacıyla residü temasları tahmin edilmiştir. Bu tahminler için residülerin fiziksel ve kimyasal özellikleri, ve birincil sekanstaki komşuları kullanılmıştır. Sonuç olarak, birbiriyle temas eden residülerin % 11'i doğru, temas etmeyen residüların % 2'si yanlış tahmin edilmiştir, ve rastlantısal bir tahminden 7 kat daha iyi sonuçlar elde edilmiştir.

İkinci bölümde, bir proteini, temas eden residülarından oluşan bir ağ olarak modellemek için yeni bir yöntem geliştirilmiştir. Bu ağların yapısal özelliklerini anlayabilmek için küçük-dünyalar fikri kullanılmıştır. Gösterilmektedir ki, residüler proteinler içinde ne düzgün ne de rastlantısal bir şekilde organizedir, küçük-dünya ağlarına benzer bir organizasyona sahiptirler. Böyle bir yapı, proteinleri çok kısa zamanlar dahilinde büyük yapısal değişimler geçirebilmesini olanaklı kılmaktadır. Ayrıca, residü ağlarının komşu sayısı dağılımları da kesik ölçeksiz dağılımlar şeklindedir. Bu da proteinlerin çok az sayıda yapısal hassas noktalar içerdiğini göstermektedir. Proteinlerin evrim sürecinde sayısız biyolojik işlevi gerçekleştirebilecek şekildeki değişimleri bu sonucu desteklemektedir. Bunun nedeni,, az sayıda hassas noktanın varlığı küçük DNA mutasyonlarının proteinlerinin yapısına yansımasına olanak sağlamasıdır.

ACKNOWLEDGEMENTS

I owe my deepest and sincere thanks to Assoc. Prof. Canan Baysal. I consider myself lucky to have met such an intelligent and dynamic person at the right time in my education. She contributed a great deal of time and energy to this thesis and has become a constant source of advice, support and love.

Special thanks goes to my dearest, Murat Kaymaz, whose love and friendship have been essential for my success.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

C_α	Central carbon atom attached to a hydrogen, an amino group, a carboxyl group and the side chain group in an amino acid
C_β	Side chain carbon atom bonded to C_α atom of a residue
Å	Angstrom

LIST OF ABBREVIATIONS

<A>	Average accuracy of a neural network
All pr.	All proteins in the dataset
C	Clustering density
C. elegans	Caenorhabditis elegans
CC	Correctly predicted contacting residues by the neural network
COF	A protein dataset comprising 225 proteins
COLD	Constrained optimization with limited deviations
DT	Delaunay Triangulation
FP	Non-contacting residues predicted as contacts by the neural network
HOT	Highly optimized tolerance
L	Characteristic path length
LRN	A protein dataset comprising 196 proteins
N1	Neural network 1 architecture
N2	Neural network 2 architecture
N3	Neural network 3 architecture
N4	Neural network 4 architecture
NN	Artificial neural network
R	Improvement of the prediction over a random predictor
TS97	A protein dataset comprising 176 proteins
WWW	World Wide Web

1. INTRODUCTION

All biological processes require different kinds of protein molecules and biological activity of any protein is achieved by its folded structure. A protein is a very complex biological macromolecule; its primary sequence governs its folding in the cellular environment and this folded state performs enormous kinds of processes such as storage, transport, catalysis, etc. Today, the major problem in biological sciences is to understand the hidden mechanisms or forces intrinsic to the primary sequence that govern the protein folding process. The answer to this question is a breakpoint for life sciences since it will enable us to design specific biological machineries to carry out specific tasks in biological cells. People from different backgrounds with different methodologies are trying to solve the folding puzzle, but no satisfactory answers could be obtained up to this point. Yet, every study contributes to the solution in various ways and helps upcoming studies to develop new ideas or strategies. In the first part of this thesis, we attempt to contribute to the solution by trying to find the contacting residues in the folded state of proteins using neural networks (NNs). The major contribution in this study is that the physical and chemical properties of amino acids are also used to predict the contacting residues in addition to the properties in previous work.

Proteins are designed to bind every conceivable molecule in the cell, from simple ions to large complex molecules like fats, sugars, nucleic acids or other proteins. They function efficiently and under control in the cells by changing their structural conformations upon binding or releasing another molecule. Therefore, resolving the structural features of proteins is an important step towards understanding structure-function duality. Proteins should be flexible enough to undergo fast and accurate conformational changes to perform their functions and this flexibility is mediated by the concerted actions of residues located at different regions of the protein [1]. Some residues play the key role during these communications and without these residues the protein would be misfunctional or nonfunctional. In the second part of this thesis,

proteins are analyzed as if they are networks of interacting residues in their folded state. We try to classify the networks of interacting residues and derive key properties of protein structure. Also, we try to determine topological characteristics of residues of a protein in three-dimensional space. The proteins are modeled as networks because (i) structure affects function in all types of networks, and this is also valid for proteins; and (ii) certain network models display a fast information relay between their nodes as well as tolerance to random failures of one or more of the nodes; these are also very important features for the functionality of proteins. Proteins need fast information relay between their residues using interacting residues in the folded state rather than their primary sequence, since they perform their functions on short time scales as low as femtoseconds. They also need to be tolerant to continuous attacks coming from the crowded environment of the cell, which may make some residue interactions impossible. Some mild residue substitutions can also be tolerated by the protein.

2. PREDICTION OF CONTACTING RESIDUES IN PROTEINS USING NEURAL NETWORKS

In this part of the thesis, a number of NNs are designed to predict the contacting residues in proteins and their performances are presented.

2.1 Overview

In order for a protein to be functional, it has to be correctly folded into its tertiary structure. In the folding process, there is interplay of non-covalent and entropic effects of the protein main chain and side chains. The folded structure of the protein have a marginal stability at its physiological conditions [2]. The hydrophobic effect is widely regarded as the major force driving protein folding. This is the energetic preference of non-polar atoms to associate and reduce their contact with water. So, the protein folds in water in such a way that hydrophobic (or nonpolar) side chains are buried inside and protected from water by water-loving (hydrophilic or polar) side-chains that make hydrogen bonding with water on the surface of the protein. Atomic packing and conformational entropy of the proteins are also important in the folding process.

The factors process mentioned above lead to a compact protein that lacks a specific architecture. The specificity of the folded structure is mediated by the hydrogen-bonding and ion pairing groups within the protein. The protein core is closely packed and it consists of non-polar and polar residues making necessary hydrogen-bonding and ion pair requirements leading to balanced charges. Unbalanced charged residues, on the other hand, are rarely fully buried. Also, exposed protein surface consists of about one-third of non-polar residues and the remaining polar atoms interact

with one another or with solvent. Disulfide bridges and salt bridges are important interactions which provide the stabilization of the folded structure [2].

Thus, in the folded state of a protein, there are specific interactions between the residues that shape its tertiary structure. These interactions could occur between two charged side chains to balance their charges in the buried space or on the surface of the protein. Hydrophobic residues can have attractive or repulsive van der Waals interactions between them that are also important for the details of the structure. In other words, if two residues are near each other, due to any of the above mentioned reasons or their combinations, less than a specific distance in the folded state, then they are called contacting residues. The contacting residues are determined by a number of strategies. One method takes all the heavy atoms of residue of interest (except its hydrogens) and draws a hypothetical sphere of a specific radius around each of the heavy atoms. If any heavy atoms of a residue are within the sphere of heavy atoms of another residue, then they are assumed to be in contact. In another method, a hypothetical sphere of specific radius is drawn around C_β atoms of each residue (C_α atom for glycine), residues having their C_β (or C_α) atoms within each other's spheres are assumed as contact. The selection of the radius of the sphere, which is called the cutoff radius, is very crucial for the specificity and non-degeneracy of the selected contacts. As the cutoff distance increase, so does the probability of having non-specific contacts. So, an optimal cutoff radius should be selected which is only large enough to select contacts of interacting residues. Another factor is that the peptide bond length is approximately is 4.5 Å, which means that adjacent residues will be in contact selecting a cutoff radius smaller than or equal 4.5 Å. So, it may be necessary to exclude these non-specific contacts coming from connectivity.

There are two main types of contacts according to relative position of the residues in the primary chain. **Short-range** contacts are the ones between the near residues in the primary sequence and they are mainly occurring within the alpha helices, beta-turns and closed loops. **Long-range** contacts are between distant residues in the primary sequence and they are occurring within the beta sheets and secondary structure elements closer in the space. Importantly, knowing the long-range contacting residues within a folded protein provides structural constraints and gives important clues about the structure of the protein.

All the contacting residues within a protein can be represented in a symmetric square matrix with size of square of the length of the protein, which is called a **contact**

map. In the contact map, the primary sequence of the protein is placed in both rows and columns of the matrix. If two residues are near to each other within a specific cutoff radius, then, the entry in the contact map corresponding to these two residues is 1, otherwise it is 0. All short and long-range interactions in a protein of known structure can be represented in its contact map. Also, secondary structures can easily be detected from contact maps [3]. Alpha helices appear as horizontal and vertical thick bands emerging from the main diagonal since they involve contacts between one amino acid and its four successors. Parallel or anti-parallel beta sheets are thin bands either parallel or perpendicular to the main diagonal respectively.

Here, long-range contacting residues in a protein are predicted using NNs in order to obtain structural constraints. Correctly predicted contacts in the folded state of the protein together with a correctly predicted secondary structure can give important clues for the structure of that protein i.e. the type of a fold. For example, Vendruscolo and his coworkers tried to recover the structure of proteins using contact maps [3]. They defined a contact map energy function to evaluate feasibility of a contact map in relation to the structural constraints of the protein of interest. By using this energy function, they tried to thread a contact map (or a 3D structure) onto a primary sequence of a protein. They are successful at recovering C_α atom contacts within 5 – 8 Å. This shows that two-dimensional contact map has valuable hidden information about the contacts in the 3D structure of the protein. This prediction may also be useful in de novo design of the proteins. In general, predicting the contacting residues within a protein corresponds to predicting the contact map of that protein. Previous attempts to predict the residue contacts within the proteins are summarized below;

Sander and his coworkers [4] predicted the protein contacts using multiple sequence alignments. They used the correlated mutational behavior of pairs of amino acids on the contact propensity. The mutational behavior is deduced from multiple sequence alignments. They showed that their method is better than other methods which do not include correlated mutations. They evaluate their performance by comparing their results with a random predictor which is an information-free predictor, and their improvement over a random predictor is five, in other words.

Casadio and Fariselli [5] predicted contact maps using NNs. They used several numbers of network architectures and fed each of them with different types of information. Their most successful network encodes the hydrophobicity and evolutionary information of the pair of residues and its neighbors. Our project involves

some of the features used in this study and also, our results with the results of their study will be compared since the strategies are similar and allow such a comparison. The similar parts of the studies will be mentioned throughout this thesis. They used the alignments from HSSP files [6] to encode evolutionary information. They concluded that their predictor is six times better than a random predictor.

Mohammed and his coworkers tried to mine residue contacts using local structure predictions [7]. There are thousands of protein structures in protein data base (PDB), but most of them cluster into around 700 fold-families based on their similarity. Thus, PDB offers a new paradigm to protein structure prediction by employing data mining methods like clustering, classification, association rules, hidden Markov models etc [7]. This method is based on the folding initiation sites and their propagation by using hidden Markov models. Their predictor is 5.2 times better than a random predictor.

What is missing in all of these attempts is the encoding of the physical and chemical features of the residues within proteins. In this study, it is aimed to encode such information to predict the contacts within proteins. We concentrate on pairs of residues and look for their contact propensity within a specified distance along the primary sequence for a given protein length. In the following chapter, NNs and their application to the specific problem at hand are summarized.

2.2 What Are Artificial Neural Networks?

Our brain is composed of about ten billion of neurons which are information processing units of the brain. They are specialized to receive, integrate and transmit the information. The input to a neuron is the electrical signals received from other neurons through its axons and the output of that neuron is the input of another neuron or a signal which directly causes an action somewhere in the body. The point of connection between two neurons or between a neuron and muscles or glands is called synapse. The physical and neurochemical characteristic of the synapse determines the strength and polarity of the new input signal which is to be sent to another neuron or cell. In other words, each neuron receives a number of signals from other neurons, but which signal is used at which amount in producing the response is decided by the synapses between the corresponding neurons. Figure 2.2.1 shows a simplified biological neuron.

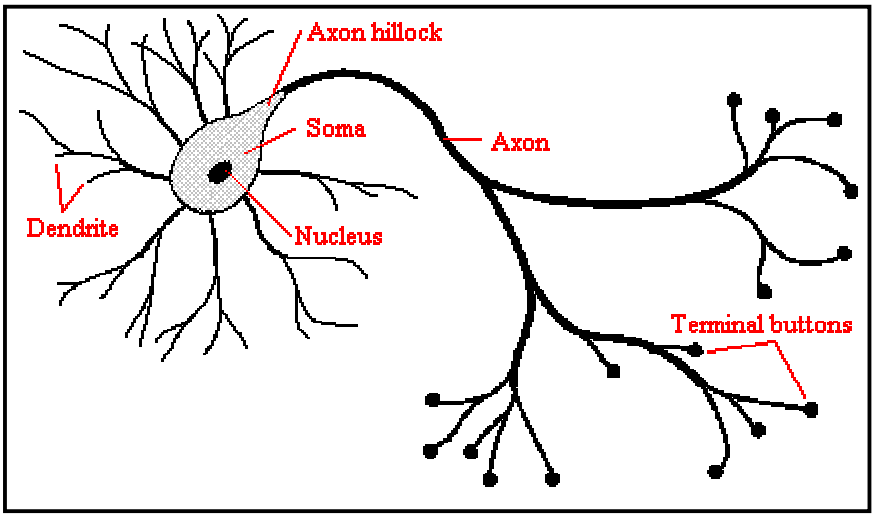


Figure 2.2.1. Schematic Representation of a Biological Neuron [8].

The brain has the capability to organize its neurons so as to perform certain computations such as pattern recognition, perception and motor control many times faster than the fastest digital computer in existence today [9]. How does the brain do this enormous computation in a very short time (on the order of milliseconds) to make us a living organism aware of his/her environment and respond to it? The answer lies within its structure which gives it the capability to build up its own rules through its *experiences*. It continuously produces or destroys connections between the neurons, and changes the type of the connections occurring within the synapses to learn and adapt to its environment.

Artificial NNs are the result of the motivation to mimic the learning and adaptation process of the brain. They are composed of simple processing units which are the artificial neurons. They learn from their environment through a learning process and connection between its units, weights, are used to store this acquired knowledge [9]. The procedure to perform the learning process is called a learning algorithm and it is defined as the modification the synaptic weights of the network to attain a desired output [9]. Figure 2.2.2 shows a simple representation of a one processing unit of an artificial NN, a neuron. In the figure, (P_1, P_2, P_3....P_n) represent a pattern. Every pattern has a corresponding target and the duty of the network is to find this corresponding output by adjusting the weights.

Input to a NN (P_1, P_2, P_3.....P_n) in Figure 2.2.2, represents a pattern by means of its appropriate features. Patterns are the examples of the problem set that needs a certain action performed on it (e.g. classification, pattern recognition etc.). For example, let's

look at training a NN that can differentiate apples from oranges. The patterns of that problem are some apples and oranges and the most suitable features to represent them would be their color and shape, because these are among their distinguishing features. It is important to note that size is not a suitable feature, since both fruits have similar size. So, the success of a network is heavily dependent upon the selection of the correct features for representing the patterns.

2.2.1 Training

There is a training phase in a NN at which the network receives a number of training patterns and adjusts its weights in order to attain corresponding outputs for each of the patterns. This phase is analogous to the time that in which the brain acquires some experiences and according to them, it makes or destroys connections between the neurons or change the nature of the synapses in order to remember and learn them. After this training process, the network is ready to test whether it can produce reasonable outputs for the patterns not encountered in the training phase, which is called generalization.

It is worth noting that weights are crude approximations to the chemical reactions occurring in neural synapses. They decide how much of the input is used in producing output as in the biological neurons.

2.2.2 Multilayer Perceptron: A NN architecture

There are many types of NN architectures and each of them has applications in different types of problems such as classification, pattern recognition, forecasting, modeling [10]. A NN type named as multilayer perceptron is very suitable for the problem in this study. Perceptron is the simplest form of a NN used for the classification of the patterns which are said linearly separable [9]. Unfortunately, many problems are not linearly separable, and they cannot be solved by a perceptron. In order to overcome this limitation, multilayer perceptrons are derived which are able to solve arbitrary classification problems.

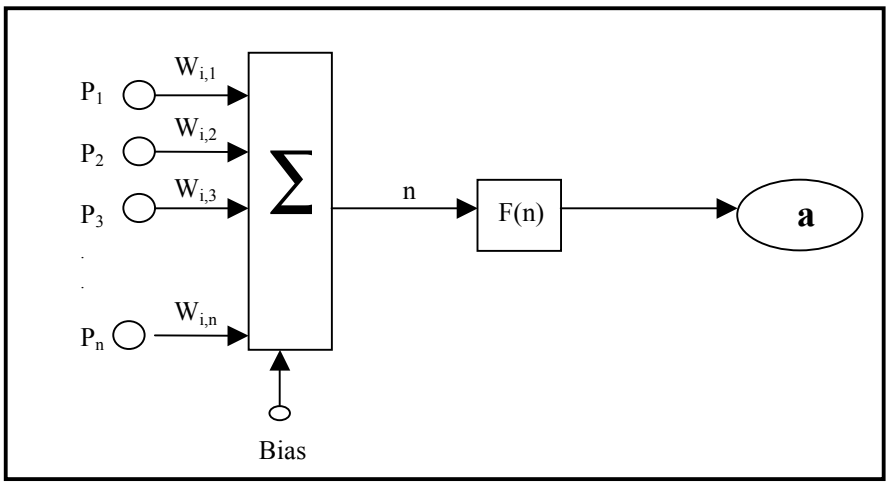


Figure 2.2.2. Simple representation of one processing unit of an artificial NN (neuron).

Bias is an optional free parameter of a neuron and it makes the network more powerful. A neuron without a bias will always gives an output of zero if the pattern features are all zero. This situation may not be desirable and can be avoided by using a bias.

To calculate the output, features of the input nodes are multiplied by the corresponding weights and the bias term is added in each summation unit (Σ) of an artificial neuron. The total input is given by;

$$n = (\sum_{j=1}^{m} P_j W_{ij}) + b_i \qquad (2.1)$$

P_j denotes the input features, W_{ij} is the corresponding weight and b_i is bias term.
The output of the neuron a is given by;

$$a = F(n) \qquad (2.2)$$

where F is the transfer function.

There are many types of transfer functions, some of them are mentioned here. In a linear transfer function, the output activity is proportional to the total input. In a threshold transfer function, the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value. In a log-sigmoid transfer function, the output varies continuously but not linearly as the input changes. Log-sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations [11]. Log-sigmoid transfer function is used in our network architectures. In this study, when a residue pair is

applied to a network, the network gives a real number output between [0, 1] interval and it denotes the contact propensity of the pair of residues applied.

Figure 2.2.3 is a representation of perceptron network architecture with one layer which means there is one set of neurons operating in parallel and producing output for each pattern.

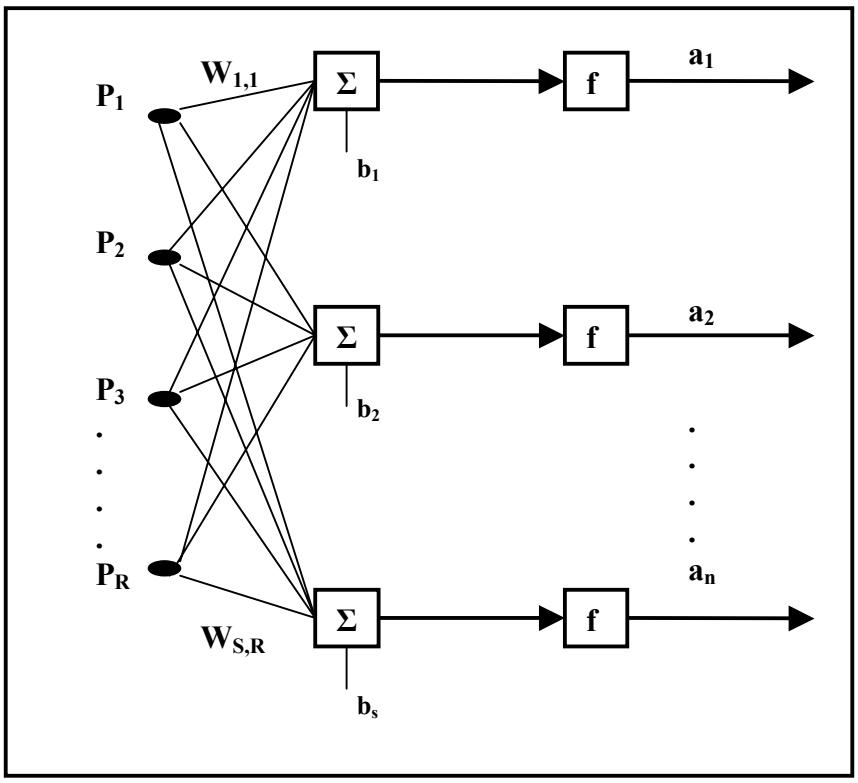


Figure 2.2.3. Layer of S number of neurons operating in parallel.

This architecture can solve only linearly separable classification problems. Linearly separable patterns mean that it is possible to classify the patterns by a line on a hyperplane as shown in Figure 2.2.4.

Multilayer perceptron architecture has evolved which can solve arbitrary classification problems including linearly inseparable pattern classification. The architecture in Figure 2.2.5 shows a two-layer perceptron. As can be seen from the figure, there are two sets of neurons operating in parallel. The nodes fed by the outputs of the first set of neurons are called hidden nodes. The number of hidden nodes varies according to the complexity of the problem.

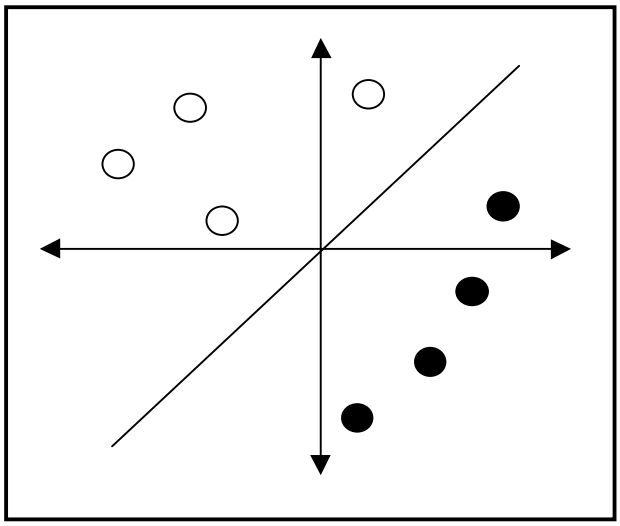


Figure 2.2.4. Patterns (white and black circles) are linearly separable

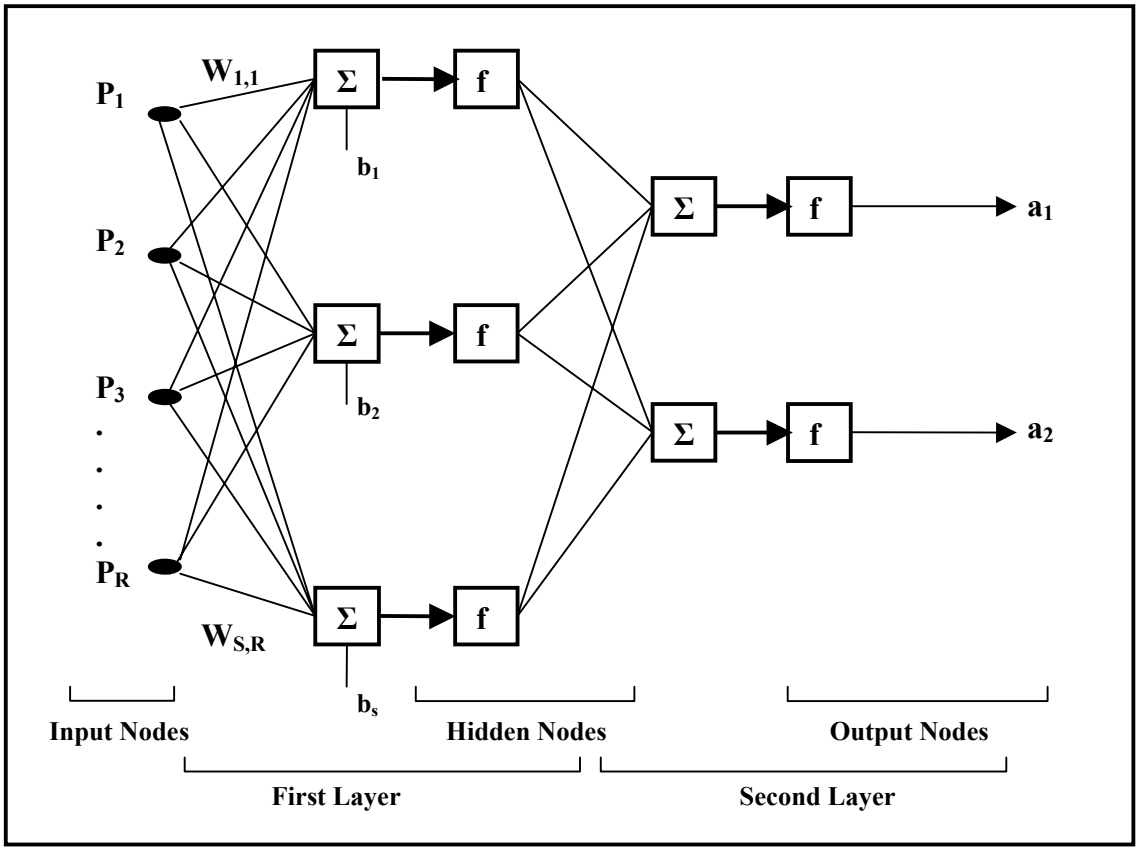


Figure 2.2.5. Multilayer perceptron architecture

2.2.3 Learning Algorithm

Learning algorithm is a procedure by which the weights and biases of a NN are modified to attain the desired output. The purpose of the learning rule is to train a network to perform a specific task. In this study, supervised learning is used. In supervised learning, there is a set of examples whose targets (correct outputs) are known, i.e. a training set. As this set is applied to the network, the network output generated for corresponding input is compared to the targets. The learning algorithm is then used to adjust weights and biases of the network in order to move the network output closer to the targets [12].

For example, in the classification of apples and oranges, the training set will be a selection of examples of apples and oranges. When a pattern in the training set (an apple or an orange) is represented to the NN, it gives an output which is the decision of the network for that pattern. This output is compared with the target which is the real class of the pattern and the weights and biases of the network are adjusted in order to move the network output towards the target. Each pattern is represented to the network and the weights and biases of the network are adjusted for each pattern. The complete representation of all the patterns in the training set to the network is called *iteration*. In order to find the appropriate weights and biases for the correct classification of all patterns, this process is iterated many times.

2.2.4 Learning and Generalization

In this project, a multilayer perceptron trained with the backpropagation algorithm is used. The essence of backpropagation algorithm is to adjust the weights and biases of the network to minimize the mean square error, where the error is the difference between the target output and the network output. Therefore, the mean square error is calculated at the end of every iteration (one pass through the set of training samples) and weights and biases are adjusted to minimize this mean square error by backpropagation algorithm. The mean square error calculated after each iteration is called training error and it tends to decrease throughout iterations. At this phase, the network learns rules in the training set and stores them in its weights and biases. Yet, there is an important trade-off in the learning process: The aim of the NN is to capture

general rules which are valid in any subset of the problem set. So, it is important to end the learning process at the correct time to prevent the over learning of the training set (generalization capacity). Therefore, in the training phase, there is another dataset, validation set, which has no common pattern with the training set. It is used to measure the generalization capacity of the network.

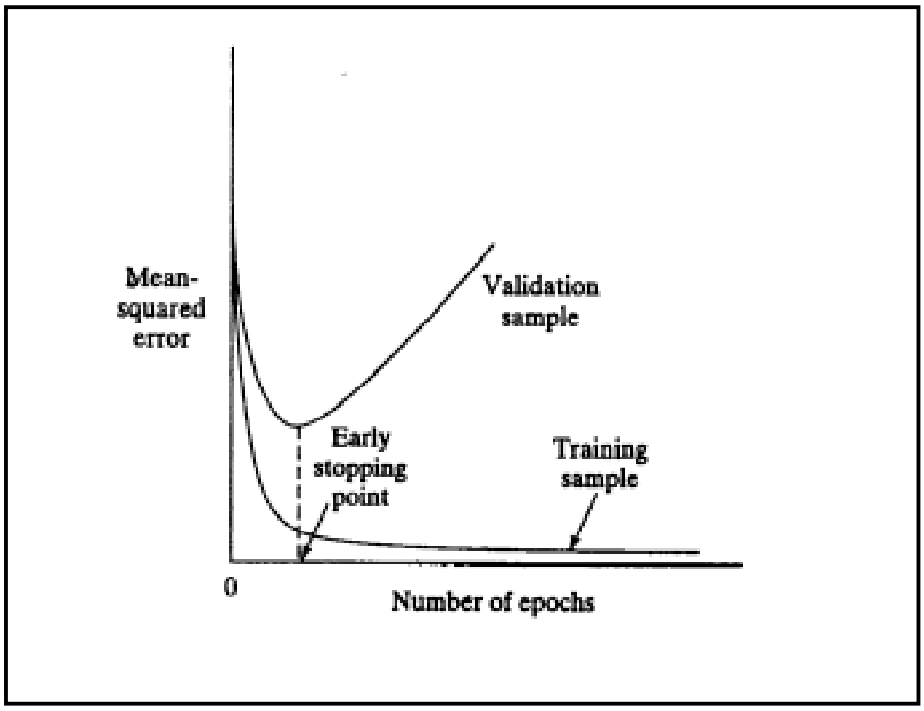


Figure 2.2.6. Mean squared error versus number of epochs in the course of training and validation phases of a typical perceptron [9]

After a set of iterations, the validation set is passed through the network and the validation error is calculated, which is the mean square error of target output and network output in the validation set. Validation and training error show a pattern like in figure 2.2.6; while the training error drops continuously, validation error increases after some time. The reason for this increase is the loosing of the generalization capacity of the network, it over-learns the training set. If the inputs used in training are a good representative of all possible input patterns, a network with enough complexity can successfully generalize what it has learned to the total population.

2.2.5 Complexity of the Network

The goal of the network is not to learn an exact representation of the training data itself, but to learn general rules from the training data which are also valid for the rest of the data. A network with enough complexity and a training dataset representative of all the dataset can achieve its goal. Complexity of the network can be considered as the number of free parameters of the network; i.e. the weights and biases. A network with little complexity gives poor generalization because of the little flexibility of the network. A very complex network relative to the problem also gives poor generalization as it fits too much of the noise on the training data [13]. In a multilayer perceptron, the complexity of the network can be adjusted by changing the number of hidden nodes since it involves changing the free parameters.

The size of the training set is an important design factor. It should be sufficient to represent the common features of the whole set of the problem. The number of iterations required for generalization is inversely proportional with the size of the training set for a network of enough complexity [9].

Several multilayer perceptrons are designed as predictors of contacting residues in proteins. These networks are trained by a backpropagation algorithm which is a supervised training method. Networks at different complexity are tried to find the optimal network architecture suitable for the prediction. In the following chapter, the problem is described and the architectures used are analyzed in detail.

2.3 Description of the Problem and the Solution Model

In this project, physical and chemical features of amino acids as well as other features involving the protein length and the primary sequence are used for predicting the contacting residues.

NNs are used for several reasons: (i) It has been shown that NNs have a very good performance on prediction problems [10]. Since our problem is also a prediction, we can safely use NNs. (ii) NNs are one of the most successful methods in protein secondary structure prediction (up to 80%) [14]. (iii) The rules determining the contacting residues in a protein are very complex. NNs are quite successful in problems

where rules crucial to the required decision are subtle or deeply hidden. NNs have the ability to discover patterns in data which are so obscure as to be imperceptible to standard statistical methods [15]. (iv) NNs have no limitations for the number of parameters in the problem to be solved. A network with enough complexity can learn as many rules as they can. Since, the number of parameters playing role in the contact decision within a protein is very high (protein secondary and tertiary structure, residue types etc.), NNs are one of the most convenient methods for a problem of this complexity.

2.3.1 Input and Output of the NN

The input of the NN is two residues or a window of residues, the length of the protein and the sequence separation of the corresponding residues (number of residues between them along the chain). The output of the network is the contact propensity of the corresponding residues. In other words, features of two residues and two other parameters are applied to the network and the desired action from the network is a prediction of these residues is in contact or not.

Three different network architectures are used in this prediction. The same network architecture is trained with different input parameters to encode more information to the network. All networks have two global parameters in common:

(i) *Normalized protein length*. Normalized length of the protein having the residue pair whose contact propensity in under examination. Normalization is achieved by dividing the length of the protein to the length of the longest protein within the whole protein set.

(ii) *Normalized sequence separation*. The number of residues between the residues of pair of interest. It is normalized by dividing it to the length of the longest protein in the whole protein set.

It is necessary to represent residues to the network by means of their specific features. Three main feature of a residue its surface area, hydrophobicity and charge are used for this purpose.

2.3.1.1 Surface Area

The area of a residue occupies in space is a measure of the size of the residue. It is strongly correlated with the size of the side chain of that residue. This feature is used to determine contact propensity of residues, because it is known that the substitution probability of an amino acid into another is inversely proportional to the difference of their sizes [16]. Sizes of the residues around the residue of interest are also important factors playing roles in their contact decision of corresponding residues. A bulky residue surrounded also by bulky residues may not be close enough to be in contact with another bulky residue which is also surrounded by bulky residues. This explains why the substitution rate between the amino acids is inversely proportional with the difference of their sizes.

Surface areas of the residues are taken from Baysal et al. study which is calculated by naccess program which is an implementation of the method Lee and Richards [17, 18] .

2.3.1.2 Hydrophobicity

It is a measure of nonpolarity of the side chains. As the nonpolarity (hydrophobicity) of the side chain increases, it avoids being in contact with water and buried within the protein nonpolar core. This is seen as the essential driving force in protein folding. This quantity is used to encode residue specific information to the network. Since the hydrophobicity of a residue affects the non-covalent bonding between its surroundings, it can be a contributing factor to contact decision of that residue with others. The hydrophobicity information can be encoded in two different ways; one method uses the hydrophobicity of the residue of interest, other method uses the average hydrophobicity of the neighbors of the residue of interest. First encoding gives only the residue-based information, tells nothing about the local environment of the residue, while the latter is giving information about the local polarity (or nonpolarity) of the environment of the residue. We calculate the average hydrophobicity according to;

$$Hyd_i = \frac{\sum_{i-3}^{i+3} Hyd_i}{7} \tag{2.3}$$

Hydrophobicity of i^{th} residue is the average of the hydrophobicities of window of residues of size seven in the primary sequence of the protein. Three of the residues that are on the left, the residue itself and three of the residues on the right of the residue constitute the window and the average of the hydrophobicities of residues in that window represent the average hydrophobicity of the residue in the middle of that window of residues. In Table 2.1, hydrophobicities of amino acids used in this prediction are listed. ROSEF hydrophobicity scale is used since it is one of frequently used scale [19, 20].

2.3.1.3 Charge

It denotes the net charge on the residue if there is any. It takes values -1, 0 and 1. Electrostatic interactions are important in determining contact propensity of the residues. Therefore, having charge feature helps to the network in learning contacts because of electrostatic interactions.

Table 2.1 shows the surface area and hydrophobicity values of 20 residues before normalization. As can be seen from the table, they are on different orders of magnitude which may not reflect their relative importance in determining the required outputs. In order to bring them on the order of unity, linear transformation is applied to the input features. Within each feature, mean and variance are calculated according to equation 3.2 and 3.3 and re-scale them according to the equation 3.4.

$$\bar{x_i} = \frac{1}{N}\sum_{n=1}^{N} x_i \tag{2.4}$$

$$\sigma_i^{\,2} = \frac{1}{N-1}\sum_{n=1}^{N} (x_i - \bar{x_i})^2 \tag{2.5}$$

$$\tilde{x_i} = \frac{x_i - \bar{x_i}}{\sigma_i} \tag{2.6}$$

where $\tilde{x}_i$ is the re-scaled variable. Hence the surface area and hydrophobicity features are re-scaled so as to be unit variance with zero mean. Normalized size and hydrophobicity and charge features of 20 residues can be seen in Table 2.2.

Residue Type	Surface Area	Hydrophobicity
ALA	107.95	0.50
ARG	238.76	-2.01
ASN	143.94	-2.26
ASP	140.39	-2.51
CYS	134.28	4.77
GLN	178.50	-2.51
GLU	172.25	-2.51
GLY	80.10	0
HIS	182.88	1.51
ILE	175.12	4.02
LEU	178.63	3.27
LYS	200.81	-5.03
MET	194.15	3.27
PHE	199.48	4.02
PRO	136.13	-2.01
SER	116.50	-1.51
THR	139.27	-0.5
TRP	249.36	3.27
TYR	212.76	1.01
VAL	151.44	3.52

Table 2.1. Surface area and hydrophobicity features before re-scaling

Each amino acid is represented by using three features, surface area, hydrophobicity and charge. This representation is aimed to correlate the physical and chemical properties of amino acids with the contact propensity. There are no previous studies for contact map prediction in which such amino acid features were used. Also, in some of the networks, the local environment of the residues is encoded in different number of ways in order to give more information to the network for prediction.

When these features are applied to the network, the output of the network is the contact propensity of the corresponding residues. It varies between 0.1 and 0.9 and 0.1 means these two residues are not contacting, 0.9 is they are in contact. But, the network gives the outputs varying from 0.1 to 0.9, so there should be a procedure which decides whether the residues are in contact or not according to the output.

Residue Type	Surface Area	Hydrophobicity	Charge
GLY	-2.00377	-0.14338	0
ALA	-1.35889	0.02916	0
SER	-1.16091	-0.66444	0
CYS	-0.7492	1.50264	0
PRO	-0.70636	-0.83698	0
THR	-0.63365	-0.31592	0
ASP	-0.60772	-1.00952	-1
ASN	-0.52552	-0.92325	0
VAL	-0.35185	1.07129	0
GLU	0.13002	-1.00952	-1
ILE	0.19648	1.24383	0
GLN	0.27474	-1.00952	0
LEU	0.27775	0.98502	0
HIS	0.37616	0.37769	0
MET	0.63713	0.98502	0
PHE	0.76055	1.24383	0
LYS	0.79134	-1.87912	1
TYR	1.06805	0.20515	0
ARG	1.6701	-0.83698	1
TRP	1.91555	0.98502	0

Table 2.2. Residue features after re-scaling. Note that charge feature is not re-scaled.

2.3.2 Contact Definition

Casadio et al. used a different contact definition that takes the distances of all heavy atoms of the residues into account and the cutoff radius is 4.5 Å. This definition is not used in this study, because being close of heavy atoms of the residues does not always mean that there is an interaction between them. The direction of the residues can be totally different but some of their atoms (for example, the backbone atoms) could still be close to each other than the cutoff radius. In order to avoid taking such non-specific contacts into account, we use only C_β atoms for contact definition. If C_β atoms of a pair of residues (C_α for glycine) are closer to each other less than 7 Å, they are assumed to be in contact; else they are assigned as non-contact.

2.3.3 Datasets

A dataset composed of 608 proteins is used for this analysis. This dataset was used before by Casadio et al. [5]. This set does not contain proteins whose backbones are interrupted. It is divided into three subsets for training, validation and test separately. Training set contains proteins without ligands in order to avoid false contacts due to the presence of hetero-atoms. Validation and test sets are composed of proteins whose sequence identity is less than 25 %. Table A in the appendix shows proteins in all three subsets with their chains.

The contacts between residues which are less than four residues apart are not included while training or testing of the networks. This type of contacts (mostly short-range contacts) is very high in number and long-range contacts are low in number respectively. So, NNs may be biased through short-range contacts because of their high number and cannot learn long-range contacts. Since our desire is to find long-range contacts in order to have a coarse structure of the protein, we exclude most of the short-range contacts.

In a protein, there are contacts much lower than non-contacts. According to our dataset and the contact definition (see section 2.3.2), the number of contacts to non-contacts ratio is 98.4. Because of this disproportion, network cannot be feed by all the residue pairs obtained from the dataset in the training phase. By doing so makes the network to output for most of the pair as non-contact, since for every contacting pair there are approximately 98 non-contacting pairs. Therefore, we have to balance this disproportion. We select all the contacting pairs generated in the training set. Then for every contacting pair, we randomly select a non-contacting pair within the dataset. Hence, a training data is prepared in which there are equal numbers of contacting and non-contacting residue pairs.

Different contact to non-contact ratio has been tried for training the networks such as 1to 2 and 1 to 6. In these cases, the network outputs have decreased dramatically and most of the pairs were classified as non-contacts by the network. So, 1 to 1 contact non-contact ratio is used for the training.

2.3.4 NN Architectures

Three different NN architectures are used to predict contacts within proteins. Each architecture differs according to information encoded in it. All the networks take a pair of residues whose contact propensity is under examination as an input and the output is the contact propensity of this pair which is a number between 0.1 and 0.9. The learning rate for all networks is 0.2 and transfer function of both hidden and output nodes are log-sigmoid which is given by;

$$\log - sigmoid(n) = \frac{1}{1+e^{-n}} \tag{2.7}$$

Since, two types of inputs are applied to one of the network architectures, four different networks are used to predict the contacting residues.

2.3.4.1 Network 1 (N1)

N1 contains eight input neurons representing the individual features of the pair of residues plus two global properties. Every feature of a residue (hydrophobicity, charge and size) is encoded by separate input neurons. Figure 2.3.1 shows the architecture of the N1. Different number of hidden nodes is used while training this network. N1 takes all the features of pair of residues and two global properties (normalized protein length and normalized sequence separation). For the sake of clarity, not all the hidden nodes and weight connections are shown in figure 2.3.1.

2.3.4.2 Network 2 (N2)

N2 has the same architecture with N1, as shown in figure 2.3.1., but it differs according to its information content. N1 takes the individual hydrophobicity of the residue while N2 takes the average hydrophobicity of the residue. Inputs to N2 are the size, charge and average hydrophobicity features of a pair of residues. Average hydrophobicity is calculated according to equation 2.3. It gives the hydrophobicity value averaged out over a window of residues which are the neighbors of the residue of

interest in the primary sequence. So, it encodes the local environment of the residue and our aim for trying N2 is to see how important this information is in the contact decision.

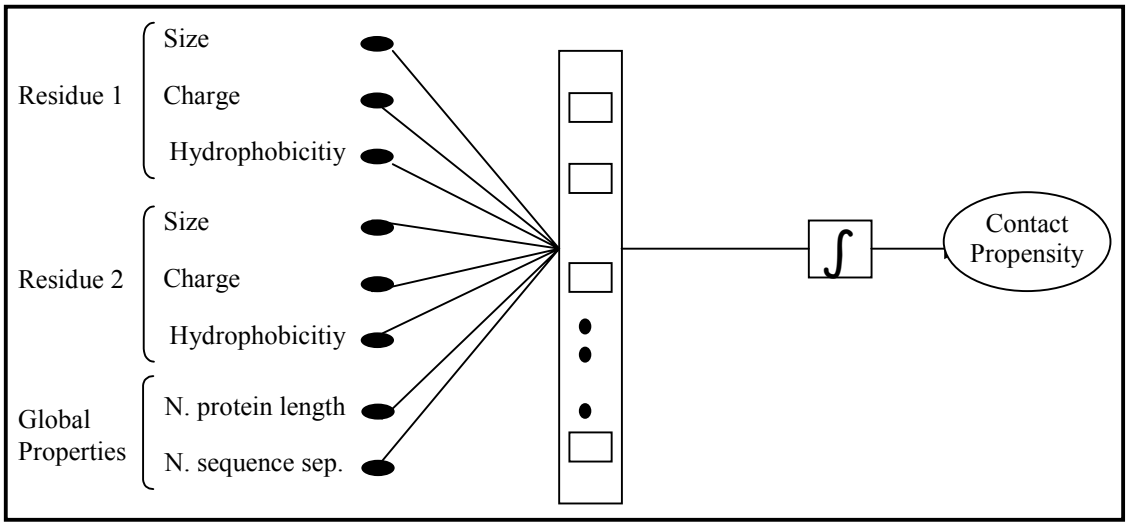


Figure 2.3.1. Architecture of N1 and N2

2.3.4.3 Network 3 (N3)

N3 has a different topology; it is very similar to the topology used by Casadio et al. in their study for predicting contact maps of proteins [5]. It contains 218 input nodes, 210 of them represents all the possible pair of residues. Each residue pair and its symmetric are encoded with the same node, which reduces the number of possible pairs from 20x20 to 20x (20+1)/2. The topology of the N3 is shown in figure 2.3.2. For the sake of clarity, not all the hidden nodes and weight connections are shown. When a residue pair is presented to N3, only one out of 210 input nodes will be 1, which is the representative of that pair, other 209 input nodes will be zero. Other 8 input nodes represent the size, charge and hydrophobicity values of the each residue in the pair of interest and two global properties (normalized protein length and sequence separation). This architecture is more complex than the previous one; it has more free variables (weights and biases) to learn the conditions of being in contact from the features of the residues presented to the network.

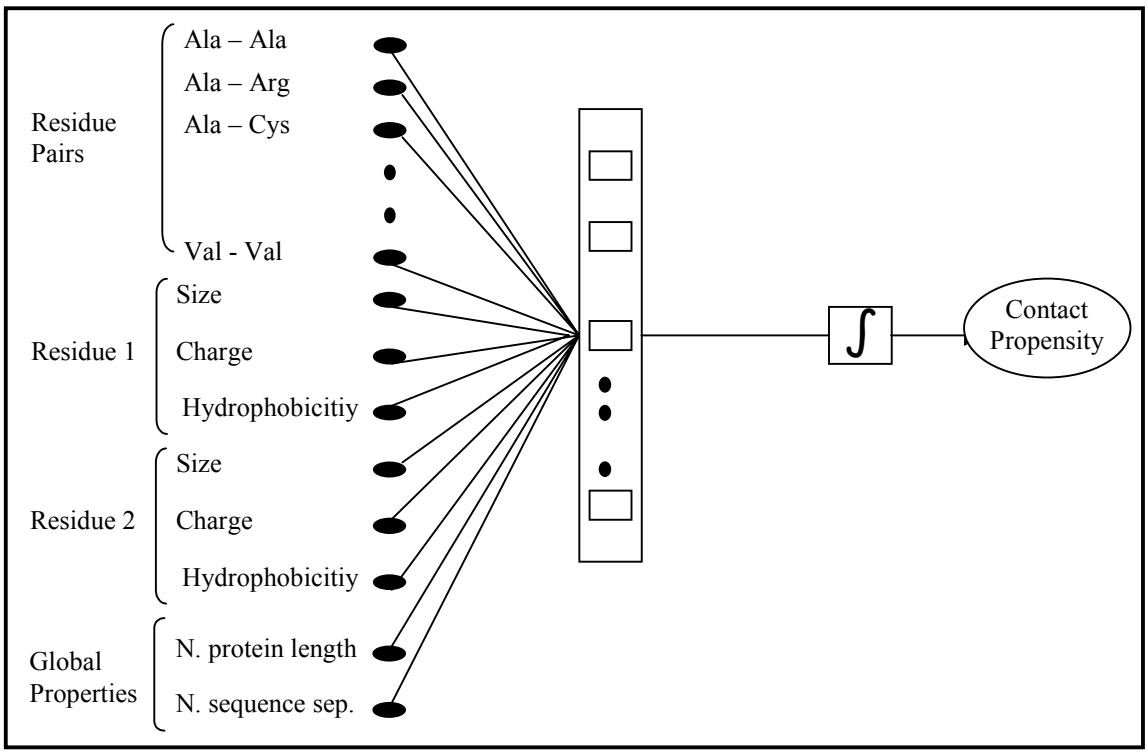


Figure 2.3.2. N3 architecture

2.3.4.4 Network 4 (N4)

In N4, a window of residues, which compromises the primary sequence neighbors of the residue and the residue itself, represents each residue. Three neighbors within the left and the right of the residue and itself constitute the window. Size, charge and hydrophobicity information of all neighbors are applied to the network. N4 topology is shown in Figure 2.3.3 and in this topology; the local environment of the residue is encoded by its neighbors in the primary sequence. In contrast to N2 where the local environment of the residues is presented by only average hydrophobicity of the neighboring residues, all the features are taken into account to represent the local environment of a residue in N4. Averaging may not be a proper way to encode the local environment, since it could not reflect the individual effects of the neighboring residues to the residue of interest. In topology of N4, effect of each neighbor is considered and an input node is assigned for each feature of the neighboring residues. So, it is a more proper way to encode local environment of the residues.

2.3.5 Evaluation of the Network Performance

In this study, two different methods are used to evaluate the network performance. In the first methodology, the number of correctly predicted contacting residues and number of false positives which are the pairs assigned by the network as contact while they are not in actual are counted. Our aim is to increase the number of correctly predicted contacts while decreasing the number false positives as much as possible. The network outputs are the real numbers in the interval of [0,1] and higher the output, more probable that the input residue pair is contacting. Therefore, in order to determine the correctly predicted contacts, we select a *threshold*. The residue pairs whose outputs are equal to or higher than selected *threshold* are assigned as contacts and other pairs are assigned as non-contacts. Correct contacts (CC) is the ratio of number of actual contacting residue pairs whose network outputs are higher than the selected *threshold* to the total number of contacts. False positive (FP) is the ratio of number of actual non-contacting residue pairs whose outputs are higher than the selected *threshold* to total number of non-contacts. They are calculated as follows;

$$\mathrm{CC} = \frac{\text{Contacting residue pairs whose network outputs} > \text{Threshold}}{\text{Total number of contacting residue pairs}}$$

$$\mathrm{FP} = \frac{\text{Non - contacting residue pairs whose network outputs} > \text{Threshold}}{\text{Total number of non - contacting residue pairs}} \tag{2.8}$$

The second method is for the comparison of the performance of our predictor with a random predictor. In this method, the network capability of predicting residue contacts is of interest [5].

Accuracy (A) of the network is defined as the ratio of the correctly predicted contacts by the network to the actual number of contacts in a protein and calculated according to;

$$A = \frac{N_c^*}{N_c} \tag{2.9}$$

N_c^* is the number of correctly predicted contacting residues by the network,
N_c is the actual number of contact within the protein.

Now, the question is how correctly predicted contacts are determined. As mentioned, the output of every network in this study is a real number in the interval [0,1] which denotes the contact propensity of the corresponding pair of residues. Higher the network output, more probable that the input residues are in contact or vice versa. So, the number of correctly predicted contacts is determined by sorting the network outputs and selecting the top outputs as much as the number of actual contacts in that protein. Correctly predicted contacts are the actual contacting pairs whose outputs are within the selected top outputs.

A random predictor makes N_c number of guess in order to predict the contacting pairs, assuming that there are N_p number of residue pairs in which N_c of them are contacting. Therefore, its performance (A_r) is calculated by;

$$A_r = \frac{N_c}{N_p} \tag{2.10}$$

Since the contact map is symmetric and residues whose sequence separation is less than four are not included, N_p is calculated by;

$$N_p = \frac{(Lp-4)\times(Lp-3)}{2} \tag{2.11}$$

where Lp is the protein length.

In order to calculate the improvement over a random predictor, accuracy A of the network is divided to performance of the random predictor A. Improvement over a random predictor is denoted by R and calculated according to,

$$R = \frac{A}{A_r} \tag{2.12}$$

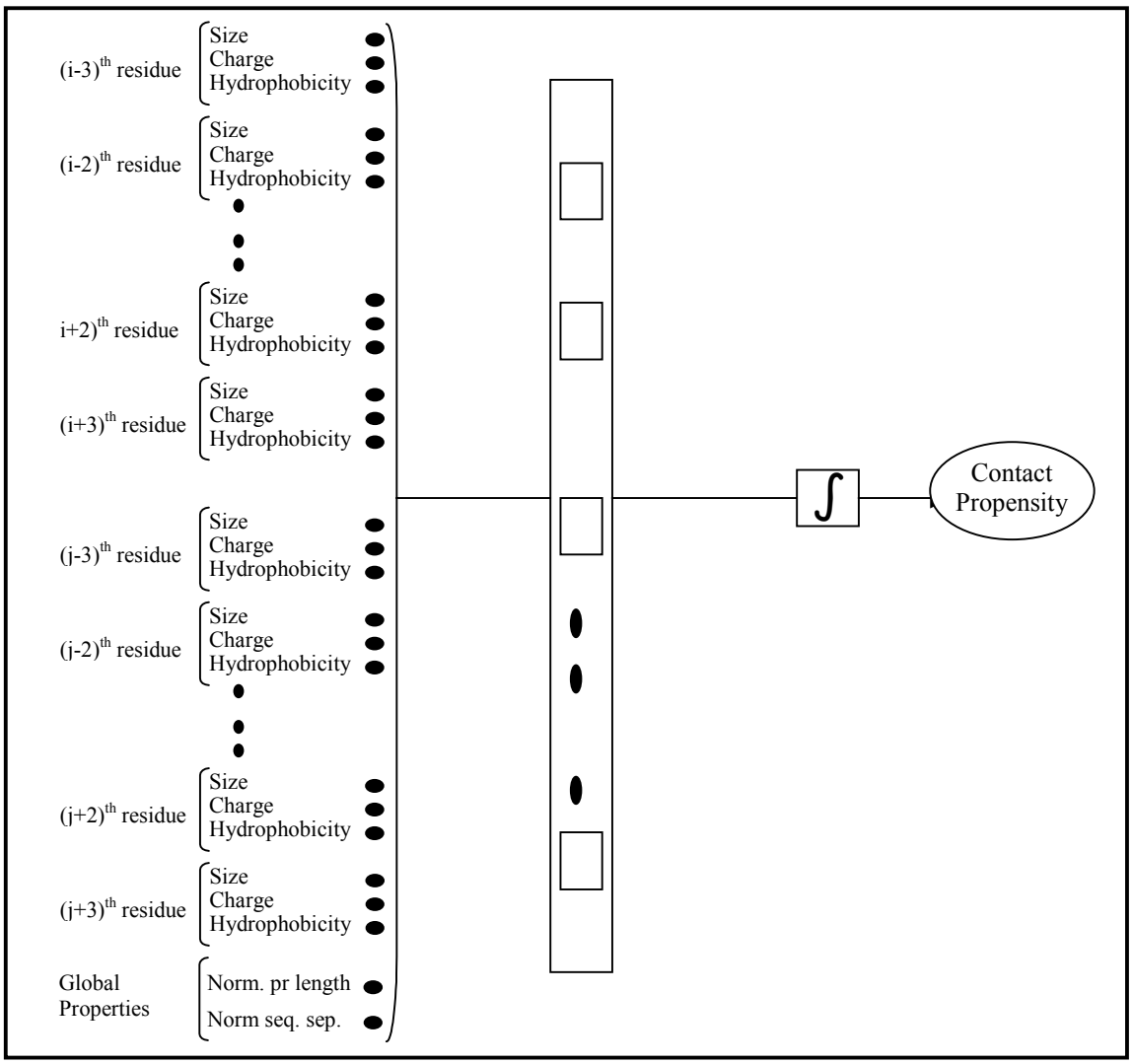


Figure 2.3.3. N4 architecture for a pair of residue *i* and *j*

2.4 Results and Discussions

All networks are trained with their corresponding training files. While training, they are tested on proteins contained in validation (TS97) dataset (see section 2.3.3). This testing is called validation and it is required to stop the training phase with up most generalization capability. Otherwise, the network will learn all the patterns in the training dataset and loose its generalization capacity over the all dataset.

The validation set is divided into four subsets according to the length of the proteins. First validation set (Val Set 1) comprises proteins whose lengths are smaller than 100 amino acids, second set (Val Set 2) comprises proteins whose lengths are

between 100 and 170 amino acids, third set (Val Set 3) comprises proteins whose lengths are between 170 and 300 amino acids and fourth set (Val Set 4) comprises proteins whose lengths are larger than 300 amino acids. The reason for this division is that the performance of any network varies significantly with the length of the protein. As protein length increases, the possible number of pairing increases with the square of protein length while the actual contact number do not varies that much. Table C in the Appendix shows the details of the proteins in the validation set (TS97). In the following experiments, network performances are calculated based on the performance on these validation datasets. All performance results are represented by a table. There are correct contacts (CC), false positives (FP), accuracy (<A>) and comparison with the random predictor (R) as explained in section 2.3.5.

2.4.1 Experiment 1

In this experiment, N1 is trained with different number of hidden nodes and the performance of it on the validation set is determined. In the training set, there are 128862 patterns whose half of them is the contacting residues and other half is the non-contacting residues which are selected randomly from the training dataset. N1 is trained using 3 different numbers of hidden nodes; 10, 15 and 20. To recall, there are 8 input nodes in the N1, six of them represent the size, charge and hydrophobicity features of residue pair of interest plus two global properties. The performances of the N1 with different number of hidden nodes are shown in Table 2.3.

	N1 with 10 hidden nodes				N1 with 15 hidden nodes				N1 with 20 hidden nodes			
	CC (%)	FP (%)	<A>	R	CC (%)	FP (%)	<A>	R	CC (%)	FP (%)	<A>	R
Val Set 1	7.1	0.8	0.151± 0.002	4.79	7	0.007	0.140± 0.001	4.42	10	0.014	0.150± 0.001	4.68
Val Set 2	1.7	0.2	0.091± 0.002	4.71	2.3	0.003	0.094± 0.002	4.77	6.0	0.008	0.125± 0.001	5.37
Val Set 3	4.1	0.6	0.074± 0.001	5.85	3.9	0.005	0.074± 0.001	5.93	6.3	0.009	0.074± 0.001	6.00
Val Set 4	7.3	0.9	0.067± 0.001	8.35	10	0.012	0.071± 0.001	9.04	9.2	0.010	0.072± 0.001	9.04
All pr.	5.8	0.8	0.083±0.002	6.35	7.4	0.010	0.084± 0.001	6.51	**7.9**	**0.010**	**0.085± 0.001**	**6.58**

Table 2.3. Performance of N1 on the validation dataset

Note that, there are two ways to evaluate the network performance, one method is to count correctly predicted contacts with the false positive ratio, and other method is to compare it with a random predictor. Table 2.3 shows performances calculated by these two evaluation methods. Training of N1 with 10 hidden nodes stopped at 6000th iteration, N1 with 15 hidden nodes stopped at 31250thiteration and N1 with 20 hidden nodes stopped at 25500th iteration.

N1 is the simplest network in our system according to the information encoded within the network. Input nodes of it encodes two global properties plus individual size, charge and hydrophobicity values of residues of interest whose contact propensity is under examination.

2.4.2 Experiment 2

It is known that the local environment of the residues influences contact decision of two residues in proteins significantly. So, in order to mimic this influence, an average hydrophobicity term is used. This method is used by Casadio et al. [5] and they use only this entity to represent the hydrophobicities of residues. In this experiment, the same network architecture as N1 is used, but, average hydrophobicities of residues are used to encode the hydrophobicities of residues in residue pairs of interest. This network is called N2 (see section 2.3.4.2.). In the training set, there are 128862 patterns in which the numbers of contact and non-contacting pairs are equal. Performance of N2 on the validation set is shown in Table 2.4. Again, validation set is divided into 4 subsets according to the protein lengths. Training of N2 with 10 hidden nodes stopped at 3550th iteration, N1 with 15 hidden nodes stopped at 103000th iteration and N1 with 20 hidden nodes stopped at 57000th iteration.

Since, N1 and N2 have the same network architecture but different information content (differing by their hydrophobicity encoding), it is appropriate to compare their performance in order to understand which hydrophobicity encoding is meaningful. N1 with 20 hidden nodes is performed best among all N1 and N2 architectures with different hidden nodes. Generally, N1 performs better than N2. Since the only difference between these two networks is the encoding of hydrophobicities, it can be said that N1 hydrophobicity encoding is more successful that that of N2. It is for sure that local environment is very important for the contact decision of residues. Based on

this, since N2 has performed poorer, it is concluded that taking arithmetic average of hydrophobicities of neighbors of residue of interest cannot represent the local hydrophobicity of that residue. Individual hydrophobicities of residues of interest are more informative for encoding the hydrophobicity.

	N2 with 10 hidden nodes				N2 with 15 hidden nodes				N2 with 20 hidden nodes			
	CC (%)	FP (%)	<A>	R	CC (%)	FP (%)	<A>	R	CC (%)	FP (%)	<A>	R
Val Set 1	0.4	0.1	0.101± 0.001	3.23	0.6	0.1	0.098±0.001	3.18	0.1	0.1	0.097±0.001	3.11
Val Set 2	0.6	0.1	0.089± 0.002	4.39	0.4	0.1	0.089±0.001	4.75	0.4	0.1	0.086±0.002	4.26
Val Set 3	3.1	0.6	0.071± 0.001	5.55	2.4	0.4	0.071±0.001	5.54	1.8	0.3	0.067±0.001	5.23
Val Set 4	6.7	0.8	0.068± 0.001	8.69	8.9	1.0	0.068±0.001	8.74	14.0	3.1	0.065±0.001	8.18
All pr.	4.9	0.7	0.077±0.001	6.08	**6.0**	**0.8**	**0.077±0.001**	**6.19**	9.0	2.4	0.074±0.001	5.77

Table 2.4. Performance of N2 on the validation dataset

2.4.3 Experiment 3

In the previous experiment, it is concluded that individual hydrophobicity is more informative than average hydrophobicity. Casadio et al. used average hydrophobicity to encode the hydrophobicities of residues in very different network structures than we use in the previous experiments; one input node is set for each possible residue pair. Their network architecture is mimicked by representing each possible residue pair with one input node. Additionally, there are six input nodes representing size, charge and individual hydrophobicities of residues in the pair of interest and two global properties. This network is called N3 and trained this network with the same training set used for N1 and 2 by using 5 and 10 hidden nodes. The same validation set is also used to stop the training with up most generalization capacity. Training of N3 with 5 hidden nodes is stopped after 100 iterations, and training of N3 with 10 hidden nodes is stopped after 400 iterations. Performances of the networks with different number of hidden nodes are shown in table 2.5.

As can be seen from table 2.5, N3 performs poorer than N1. Based on the *R* score (the improvement over a random predictor), N3 architecture performs nearly the same as the networks presented in Casadio et al. study in which their results are six times better than a random predictor (*R*=6). To understand how our additional features

improve the network prediction capability, we need to look at the simplest network model in Casadio study, which have 210 nodes representing each possible pair, 2 nodes representing average hydrophobicities of residues of interest plus 2 global properties. This network performs 5.5 times better than a random predictor over the same validation set. Therefore, using size and charge of residues and using individual hydrophobicity instead of average hydrophobicity enable the network performs better.

	N3 with 5 hidden nodes				N3 with 10 hidden nodes			
	CC (%)	FP (%)	<A>	R	CC (%)	FP (%)	<A>	R
Val Set 1	6.2	0.6	0.146± 0.001	4.63	7.6	0.8	0.147± 0.001	4.56
Val Set 2	3.0	0.5	0.091± 0.002	4.50	2.9	0.4	0.091±0.002	4.58
Val Set 3	4.0	0.6	0.072± 0.001	5.67	4.1	0.6	0.073±0.001	5.76
Val Set 4	8.2	0.9	0.069± 0.001	8.67	6.0	0.6	0.066±0.001	8.17
All pr.	4.2	0.5	0.081±0.001	6.10	**5.1**	**0.6**	**0.081±0.001**	**6.13**

Table 2.5. Performance of N3 on the validation dataset

2.4.4 Experiment 4

In this experiment, the performance of N4 on TS97 validation set is investigated. N4 uses size, charge and individual hydrophobicity information of neighbors of residue of interest. Neighbors of a residue are represented by a window of residues compromising residues whose three of them is on the left and three of them on the right in the primary sequence and the residue itself (see section 2.3.4.4.). With this encoding, it is aimed to represent the local environment of a residue in a much better way. Table 2.6 shows the performance of N4 with different number of hidden nodes on validation set TS97. Training of N4 with 15 hidden nodes stopped at 9000^{th} iteration, training of N4 with 20 hidden nodes stopped at 10500^{th} iteration.

N4 with 15 different hidden nodes performs better than all other networks in this and previous experiment. Its improvement over a random predictor is 6.75, which is also higher than the improvement of the networks in Casadio study. By looking at these results, it can be concluded that a better way is found to represent the local environment of the residues. Instead of averaging of the hydrophobicities, separate input nodes are

assigned for each different feature of neighboring residues. This also proves again that, local environment of the residues is effecting their contact decision; otherwise, such an improvement cannot be seen.

	N4 with 15 hidden nodes				N4 with 20 hidden nodes			
	CC (%)	FP (%)	<A>	R	CC (%)	FP (%)	<A>	R
Val Set 1	13.8	2.7	0.143± 0.003	4.19	10.5	2.4	0.123± 0.002	3.50
Val Set 2	8.1	1.6	0.098± 0.002	4.64	9.3	1.9	0.096±0.002	4.51
Val Set 3	9.4	0.2	0.082± 0.001	6.21	10.8	2.0	0.081±0.001	6.19
Val Set 4	14.1	0.2	0.076± 0.001	9.64	12.9	1.6	0.073±0.001	9.23
All pr.	12.1	1.9	0.089±0.002	6.75	**11.9**	**1.7**	**0.086±0.001**	**6.51**

Table 2.6. Performance of N4 on the validation dataset

With these results, which networks are tested on the test dataset (COF) is determined. The best networks in each experiment are selected and tested on the test dataset.

2.4.5 Test Results

In the previous sections, a number of networks are trained with the contact and non-contact information collected from a set of proteins and these networks are tested on another set of proteins to stop the training at up most generalization capability of the networks so to set the weights of the networks. Now, a different set of proteins is used to test these best performed networks whose weights are set. This testing enables us to measure the performance of the networks for prediction of contacts in proteins. Protein data set which is used for this testing is COF dataset (see section 2.3.3.).

Four different NNs are designed to predict the contacting residues in proteins and the best performed architectures and weights are chosen from each type of the network. N1 which is the simplest network is best performed with 20 hidden nodes; hence that network is tested on COF dataset. Similarly, N2 is best performed with 15 hidden nodes, N3 with 10 hidden nodes and N4 with 15 hidden nodes. Table 2.7 shows the performances of these networks with set weights on proteins with different sizes and on

all proteins in the test dataset. N4 with 15 hidden is the most successful network on the test dataset. In order to assign an average accuracy and improvement over a random predictor, the performances of N4 is averaged out over validation and test datasets. Therefore, this predictor has an average 0.086 accuracy and 7 times better than a random predictor.

	N1 with 20 hidden nodes				N2 with 15 hidden nodes				N3 with 10 hidden nodes				N4 with 15 hidden nodes			
	CC (%)	FP (%)	<A>	R	CC (%)	FP (%)	<A>	R	CC (%)	FP (%)	<A>	R	CC (%)	FP (%)	<A>	R
Val Set 1	10.9	1.7	0.156±0.001	3.99	1.5	0.2	0.086±0.004	1.90	9.0	0.8	0.148±0.001	3.83	12.6	3.6	0.160±0.003	3.52
ValSet 2	4.5	0.7	0.085±0.001	4.17	0.3	0.1	0.080±0.001	3.99	2.2	0.4	0.081±0.001	3.92	7.3	1.3	0.094±0.002	4.52
ValSet 3	5.9	0.9	0.076±0.001	5.63	2.5	0.4	0.075±0.001	5.63	3.7	0.6	0.072±0.001	5.30	9.2	1.5	0.081±0.001	5.94
ValSet 4	9.5	1.0	0.070±0.001	9.14	9.3	1.0	0.071±0.001	9.33	6.1	0.6	0.066±0.001	8.50	13.3	1.7	0.078±0.02	9.85
All pr's	8.1	1.0	0.078±0.001	6.68	6.8	0.8	0.0075±0.001	6.64	5.2	0.6	0.074±0.001	6.23	**10.8**	**1.7**	**0.086±0.031**	**7.03**

Table 2.7. The performances of the best networks on the test dataset COF

3. PROTEINS AS NETWORKS OF THEIR INTERACTING RESIDUES

In this part of the thesis, proteins are analyzed as networks of their interacting residues. Small-world network concept is used to explain the characteristics of resulting residue networks.

3.1 Overview

All biological processes require different kinds of protein molecules and the biological activity of any protein is performed by the folded structure of that protein. At physiological temperatures, folded proteins have conformational flexibility that is essential for their biological activities. This flexibility is mediated by the concerted actions of residues located at different regions of the protein [1, 21, 22]. Some residues play a key role during these communications and without these residues the protein would be non-functional [23].

In this project, structures of folded proteins are analyzed considering them as networks. A network is a collection of nodes which are partially or fully connected to each other. A node can be any entity such as a substrate in a metabolic network, a station in a subway network or a neuron cell in our brain. We are surrounded by an enormous number of small or large-scale networks in our real life. In any network, the collection of connections between its nodes, which are called edges, give the structural (topological) characteristic of the network.

Since networks are everywhere in our lives, understanding the efficiency, speed and accuracy of the networks is crucial. On 10 August 1996, a fault in two power lines in Oregon, USA led, through a cascading series of failures, to blackouts in 11 US states and two Canadian provinces, leaving about 7 million customers without power for up to

16 hours [24]. A computer virus named as Love Bug worm is the worst computer attack to date; it spread over the Internet on 4 May 2000 and inflicted billions of dollars worth of damage worldwide [24]. In these examples of failed networks, topology (connectivity) and its fragility to random failures play a major role.

Structure of a network always affects its function. The speed, accuracy and efficiency of a network are determined by its topology, in other words, structure of the network. This sounds very familiar to protein scientists, since the structure of a folded protein affects the function of that protein. This partly constitutes our motivation to examine folded proteins as networks. A protein is converted into a network of its interacting residues by using the Cartesian coordinates of its residues which are in turn converted to a contact map [25]. In order to separate these networks from the networks of interacting proteins, which are called protein networks, we name them as residue networks. From this point on, a residue network refers to a folded protein converted into a network of its interacting residues. Then, a number of network parameters is calculated from residue networks. These parameters help us analyze the structure of a protein as a network of its interacting residues.

A network can be regular, random or between these two extremes according to the mode of its connectivity. Real-life networks that have evolved naturally are neither regular nor random, but they have characteristics between these two extremes. In order to understand the connectivity of the network, two main parameters are required; **characteristic path length** (*L*) and **clustering density** (*C*). L is the average of the shortest paths required to go from one node to another node within a network. It gives an idea about the diameter of the network. *C* is a measure of local clustering within a network. If a number of nodes are highly inter-connected to each other, then there is a high probability to have a cluster in that *location* of the network. It is a local property and it gives an idea about the attack tolerance of the network. A regular network has many clusters with long path lengths while random networks have shorter path lengths with a small number of clusters. A small-world network topology, on the other hand, has the best of both, with short path length and high *C*. Small-world network concept was first introduced by Watts and Strogats [26]. Networks showing small-world network behavior have short path lengths, which makes it easier to go from one node to another. They also have high *C*, which makes the network tolerant to random failures of a few numbers of nodes. It is shown that many real-life networks show these properties and they are classified as small-world networks such as the World-Wide Web [27], the

Internet backbone [28], the NN of the nematode worm Caenorhabditis Elegans [29] and metabolic networks [30].

There are previous studies to examine the packing of proteins on global and local scales. Residue packing in the protein interior has long been considered to be essential to the native-like character, stability, and function of proteins [31, 32]. Raghunathan et al. found that all residues conform almost perfectly to a simple lattice model for sphere packing when a radius of 6.5 Å is used to define non-bonded (virtual) interacting residues. Side-chain positions with respect to sequential backbone segments are relatively regular as well [32]. However, a regular network model for a protein cannot provide short distances which are required for the concerted actions of residues at different regions of the protein in very short time scales [1]. A recent study shows that packing in proteins is on average similar to random packing of hard spheres encountered in soft condensed matter [33]. Another study done by Liang et al. shows that packing in proteins behaves like random spheres near their percolation threshold [34]. Random packing of proteins would provide the short distances for the fast information relay between residues, but this cannot warrant the high clustering similar to regular packing. Thus, a special network model is required for proteins to explain their conformational flexibility together with their highly packed globular structures. Small-world networks characterized by their short path lengths and highly clustered structures are candidate topologies to explain such properties of proteins. To explore this phenomenon, a method is defined to consider a single protein as a network of its residues.

In order to analyze proteins as networks, each protein is converted into a network of its interacting residues. Figure 3.1.1 and 3.1.2 show two different representations of residue networks. In figure 3.1.1, it is a three dimensional representation of a residue network in which residues that are closer to each other than a given distance (7 Å) are connected by an edge. Connectivity is shown by black lines and adjacent neighbors are shown by gray lines. In figure 3.1.2, all residues are aligned on an ellipse for a clearer understanding of its topology and the connections between them are displayed. In this figure, there are no spatial constraints on the place of residues. These figures were drawn by a network drawing program named Pajek [35]. We calculate the L and C parameters of such residue networks for a large number of proteins. To interpret these results, they should be compared with those of random networks. For this purpose, each

residue network is randomized by keeping the number of neighbors of each residue constant, but changing the neighbors of each residue.

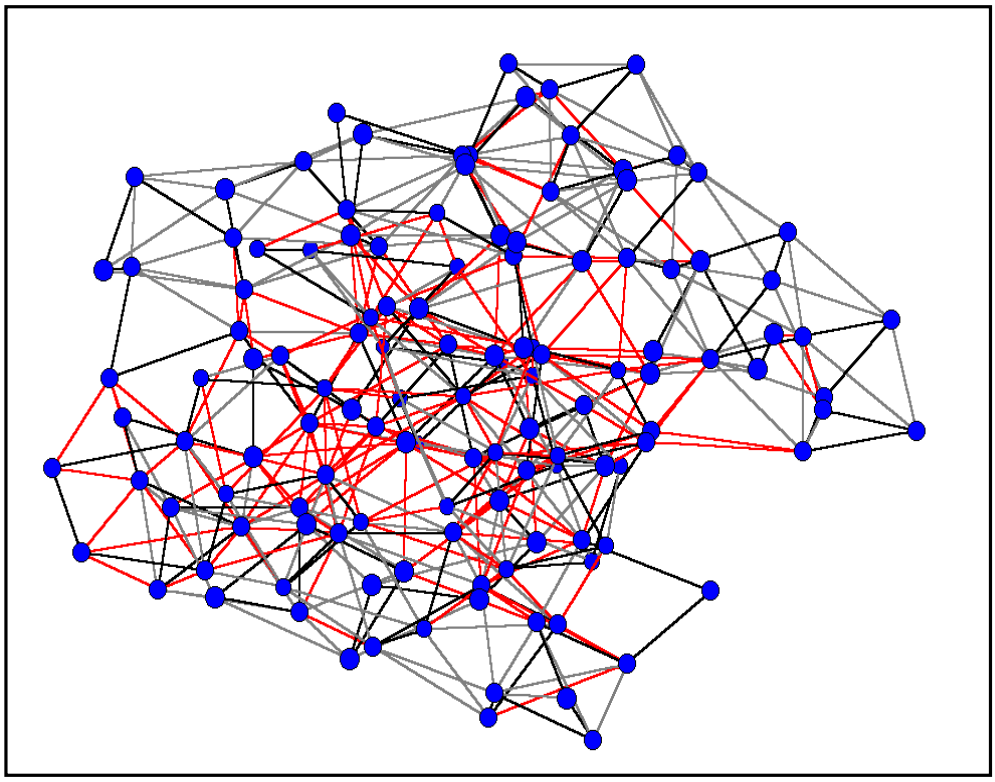


Figure 3.1.1. Residue network of 3chy protein generated at 7 Å

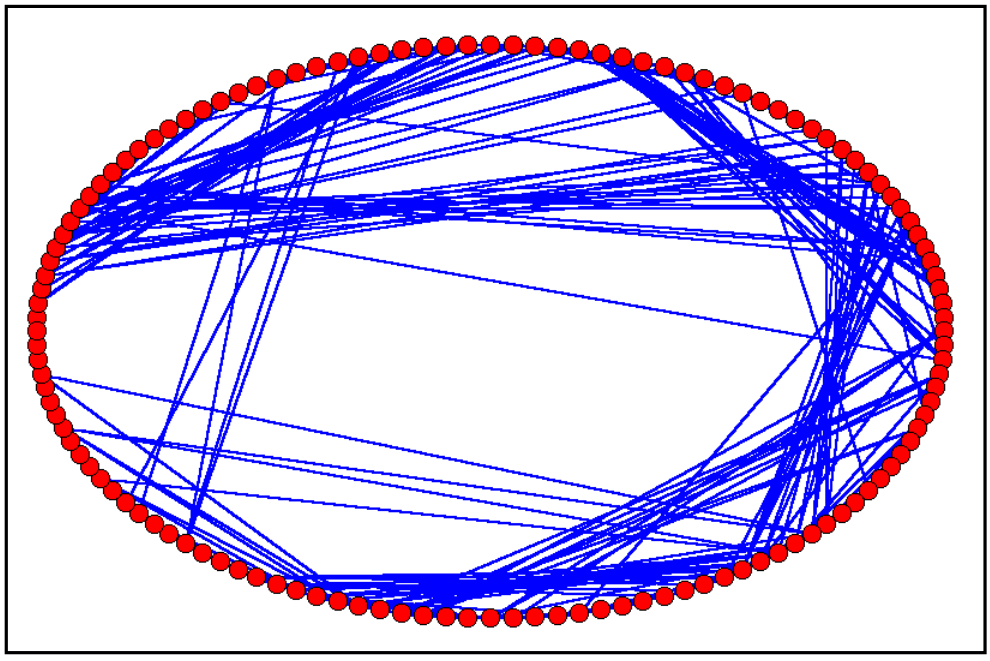


Figure 3.1.2 Another representation of 3chy residue network. No spatial constraints are used for the generation of this network.

Proteins are also converted into residue networks using Delaunay Triangulation (DT) which is a very powerful method for creating networks without having resort to a cutoff distance. Network parameters of all these networks are calculated and compared with those of actual networks in order to judge the characteristics of residue networks. Results are also compared with other parameter sets.

Another property that is crucial for determining the characteristics of a network, is the degree distribution; i.e. the distribution of the number of neighbors of all nodes in the network. It is well-known that small-world networks display different degree distribution patterns than random and regular networks [36]. Thus, degree distribution of residue networks is also calculated and analyzed to understand the characteristics of the residue networks.

Since contact maps of proteins are used to convert the latter to a network of its residues, a cutoff radius has to be chosen to determine the interacting residues. Cutoff radius is required to decide whether any pair of residues is in contact or not according to the Euclidian distance between the selected atoms of corresponding residues. Therefore, changing the cutoff radius changes the topology of the network. Because of this, network properties of proteins are analyzed as a function of the cutoff radius at which the networks are generated. The choice of a cutoff radius to form the contact map may be arbitrary at times. A procedure that will decide on the edges connecting the nodes automatically, free from a choice of cutoff radius is desirable. Such a procedure is offered by DT [37, 38]. By using DT, one can generate a protein network without resorting to any cutoff distance to determine the contacting residues, since in the triangulated protein, each edge of the tetrahedral is also an edge connecting two residues. DT has been used in proteins to understand the nature of packing and the structure. [33, 34].

3.2 A Closer Look at Small-World Networks

We live in a world of complex networks. Any complex system can be modeled as a network, where the vertices are the elements of the system and the edges represent the interactions between them [39]. The global economy is a network of national economies, the brain is a network of neurons, and metabolic networks are networks of

substrates and products of metabolic reactions. Networks that exist in real life such as WWW, food webs, Internet, protein networks in a cell are modeled and it turns out that they represent unique properties which give them flexibility, speed and error tolerance. Spreading of diseases through social networks or propagation of cascading failures through large power grid or financial systems are also mediated by the networks with unique properties mentioned above [40]. Therefore, understanding the underlying mechanisms and principles behind these efficient networks will be fruitful in a remarkable variety of fields [40] and will aid the design of more efficient networks.

Regular networks or random networks cannot explain the complex topology of the real life networks. In the toy model of Watts and Strogats [26], there is a ring lattice with n vertices and k edges per vertex as shown in figure 3.2.1. Each edge is rewired at random with probability p, and in the course of this rewiring, graph is tuned between regularity ($p = 0$) and disorder ($p = 1$). As can be seen in figure 3.2.1, the regular network is highly clustered while the random network does not have such clusters; on the other hand, the latter has much shorter path lengths than its regular counterpart. The network in the intermediate region has clusters like regular networks but shorter path lengths like random networks. These networks are called small-world networks and they are used to model and understand real-life networks.

Barabasi et al. explored the complex interaction of metabolic networks in 43 organisms [30]. They examined the topological properties of these 43 different organisms based on data in the WIT database [41]. A metabolic network is built up of nodes, the substrates, that are connected to one another through links that are the metabolic reactions [30]. They show that metabolic networks are best modeled by the small-world network model. They are robust and error-tolerant networks which indicates that removing a few substrates does not affect the average shortest biochemical pathway between the remaining nodes. Nervous system of C. elegans which comprises 282 neurons and synaptic connections between them is mapped as a network and the properties of the resulting network are analyzed. It is concluded that it is a small-world network in the sense that its average number of steps to go from one neuron to another is close to that of an equivalent random graph, yet it is highly clustered than its random counterpart [40].

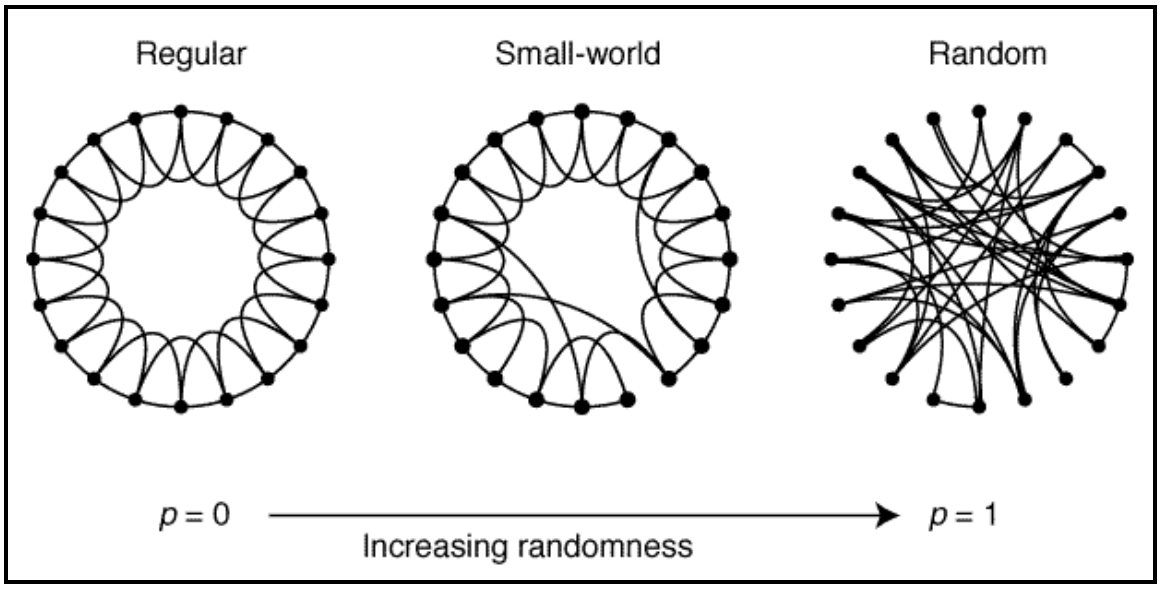


Figure 3.2.1. The transition from regular to random regime in a simple topology [26]

Figure 3.2.1 displays a regular ring lattice and its rewiring procedure with increasing randomness without altering the number of vertices or edges in the graph. A ring of n vertices, is connected to its k nearest neighbors by undirected edges. In this legend, there are 20 vertices ($n = 20$) and each of them is connected to 4 vertices ($k = 4$). In the rewiring process, a vertex and the edge that connects it to its nearest neighbor are chosen. With probability p, this edge is reconnected to a vertex chosen uniformly at random over the entire ring, with duplicate edges forbidden. This process is repeated for each vertex in the lattice. Then, the same process is repeated for more distant neighbors until each edge in the original lattice has been considered once. For $p = 0$, the original ring is unchanged; as p increases, the graph becomes increasingly disordered until for $p = 1$, all edges are rewired randomly. The claim in this figure is, for the intermediate values of p, the graph is a small-world network; highly clustered (large C) like a regular graph, yet with a small L, like a random graph [26].

To quantify these properties, there are two important network parameters playing major role in determining the overall topology of any network; L and C.

3.2.1 Characteristic Path Length (*L*)

One of the important quantities that may be calculated for networks is the *L*. It is the typical average distance between every vertex (or node) and every other vertex [40]. "Distance" does not refer to the metric space between the vertices. It refers to the minimum number of edges that must be traversed in order to reach from one vertex to another vertex; i.e. the shortest path length between the corresponding vertices [40]. It is a measure of the typical separation between two vertices in a graph.

The specific value of *L* of a network is not indicative of the topology of the corresponding network. Rather, the scaling of the *L* with the number of nodes or the average neighbor number of the nodes which is called "*L* scale" is indicative of topology of the network. As mentioned above, a network can be tuned between order (regularity) and randomness by changing the rewiring probability of each edge which is *p*. Different values of *p* represent different topologies, and the graphs with different sizes or average neighbor numbers but with the same *p* value are qualitatively same. Hence, although the *L* of a set of graphs with different sizes and average neighbor numbers, but generated with same *p* can vary over 1 to infinity, the scaling of the *L* with the number of vertices (size) or average neighbor number remains the same. This means that knowing the rewiring probability *p* of a small network enables us to obtain knowledge of its much larger cousins whose properties cannot be computed directly by using the *L* scale [40]. Also, it is a distinctive parameter of a set of networks with the same wiring probability *p*, by giving important information about their topology.

Since *L* gives the typical distance required to go from one node to another in a network [42], it also defines the diameter of the network. In regular networks, network diameter is very high and it proportionally increases with size of the network *n* ($L \sim n$) [43]. So, in large worlds, like regular networks, as the network grows, the typical distance between the nodes is linearly increasing with number of nodes *n*. Conversely, in the small-world regime, the network diameter increases proportionally to the logarithm of *n* ($L \sim ln(n)$) [43].

3.2.2 Clustering Density (*C*)

In a network, how densely the vertices are neighbors of each other is an important factor for determining the topology of the network. Clustering coefficient characterizes the extent to which vertices neighbor to any vertex are also neighbor of each other. In other words, clustering coefficient of a vertex is the measure of the inter-neighboring of the neighbors of this vertex. *C* of a network is the average of the clustering coefficients of every vertex in the network.

The clustering coefficient can be defined as follows; suppose that there is a vertex with *k* number of neighbors. There can exist a maximum of $n(n\text{-}1)/2$ edges between these *k* neighbors of that vertex. This occurs when every neighbor is connected to every other neighbor of that vertex. The ratio of the actual number of edges between the neighbors of that vertex to the maximum possible number of edges gives the clustering coefficient of that vertex. In figure 3.2.2, clustering coefficient of one vertex with three neighbors is calculated at different connectivities of its neighbors.

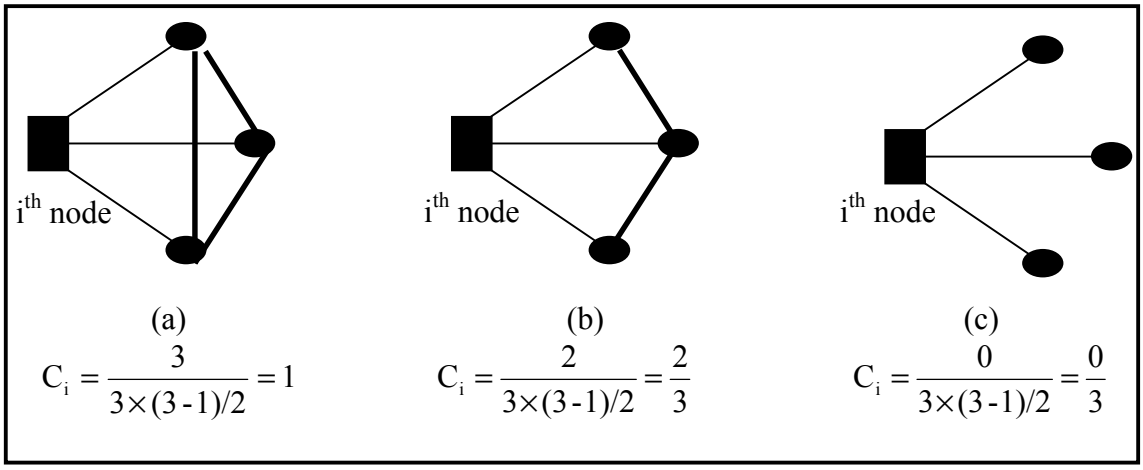


Figure 3.2.2. Calculation of clustering coefficient of i^{th} vertex in a network

Clustering coefficient of a network determines its *C*. It is an important property which gives us the information about the existence of clusters within the network. Nodes or vertices having high clustering densities are candidates for being an element of a cluster. Therefore, it is also a local property that quantifies the local regions of the network. This parameter is crucial when a network is to be compared with its random counterpart. Since random networks lack clusters, the *C* of an actual network gives an idea of how its topology differs from a random one that has a low *C*.

In order to explain how these two properties of networks are crucial for determining the topology, here are some empirical examples. Three real-life networks, film actors, power grids and neural network of C. elegans are taken into account and their network topologies are explored [26]. Two film actors are joined by an edge if they have been in a film together. This information was taken from Internet Movie Database (http:\\us.imdb.com) in April 1997. For the power grid, vertices represent generators, transforms and substations, and edges represent high-voltage transmission lines between them. For C. elegans, an edge joins two neurons if they are connected by either a synapse or a gap junction. L and clustering coefficient for all three networks are calculated and they are compared to random graphs with the same number of vertices and same average number of edges per vertex respectively. Table 3.1 shows the network parameters of three networks and their counterparts. In the film actors network, n=225,226 and k=61; Power grid, n=4,941 and k=2.67; C. elegans, n=282 and k=14. The actual Ls of the three networks are near that of their corresponding random counterparts. However, the clustering coefficients of the actual networks are much higher than their random counterparts. Therefore, all three of them show the small-world phenomenon in this perspective.

Network type	L_{actual}	L_{random}	C_{actual}	C_{random}
Film actors	3.65	2.99	0.79	0.00027
Power grid	18.7	12.4	0.080	0.005
C. elegans	2.65	2.25	0.28	0.05

Table 3.1.Examples of small-world behavior; $L \geq L_{random}$ but $C >> C_{random}$

3.2.3 Degree Distribution

In any network, number of neighbors of the nodes carries valuable information on the structure of the network. To quantify this, let $p(k)$ denote the fraction of nodes that have k links. Here k is called the degree and $p(k)$ is the degree distribution [24].

The simplest random graph models are presented by a bell-shaped Poisson degree distribution [24]. In these networks, there are no rules or preference of incoming nodes for attachment to already existing nodes, which results in a normal distribution. Real-life networks, on the other hand, are analyzed and shown as small-world present

different degree distribution patterns than randomly organized networks. Small-world networks are shown to have a large number of nodes having few neighbors besides a few number of nodes having many neighbors. Such connectivity results in a power-law which is also called scale-free distribution, since there is no single scale to define the distribution. To be scale-free is common but not universal for small-world networks [24].

The network of movie-actor collaborators, the NN of the worm C. elegans, WWW and the network of citations of scientific papers are scale-free, that is they have a distribution of connectivity that decays with a power-law tail. Scale-free networks grow in such a way that new vertices connect preferentially to the more highly connected vertices in the network; this property is absent in randomly organized networks. Hence, there are a few nodes with very high degrees dominating the topology of the networks, which are called hubs and there are many nodes having few neighbors. P(k) distribution decays as a power law $P(k) \sim k^{-\gamma}$ in scale-free networks where γ has most commonly been observed between 2.1 and 2.4. Log-log plot of scale-free distribution conforms to a line whose slope is γ. Figure 3.2.3 shows how random and scale-free organizations differ in the topology and degree distribution.

The growing character of the network is important to be free of scale or not. Barabasi et al. correlate the growth of a network model with its degree distribution as a function of time [45]. They found that network topologies at different time steps (hence having different number of nodes) growing with preferential attachment show the same degree distribution independent of time, hence the size of the network. Also, they found that networks growing without preferential attachment eliminate this scale-free distribution.

Why scale-free networks? The advantage of being scale-free is that the network is resistant to random failures, because a few hubs dominate the topology [46]. So, the probability of having attacks to the nodes having few links is higher because of the abundance of these nodes and these attacks can be easily tolerated by the topology. However, the weakness of this type of distribution is that any attack to the hubs might lead to a drastic failure of the network as mentioned in section 3.2.2.

It is understandable that many real-life networks have scale-free distributions. World Wide Web has a number of pages having high number of links and called hubs. Therefore new nodes will link also one or more of these pages in order to be reached easily by using the links of those hubs or to reach easily to other pages in the WWW.

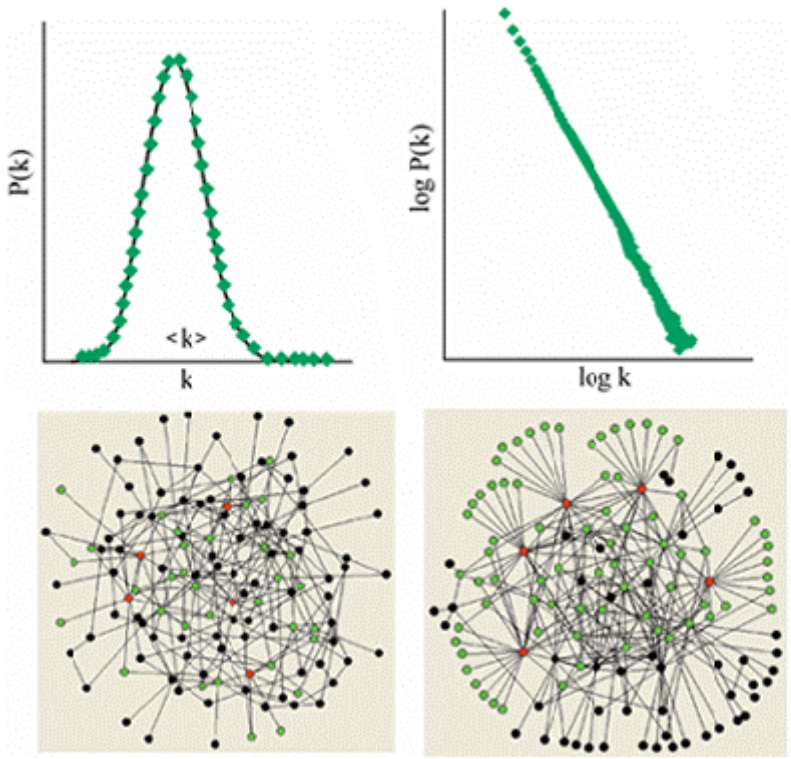


Figure 3.2.3. Degree distribution of random and small-world networks. The network on the left is randomly wired and its degree distribution presents a Gaussian distribution. The network on the right is a small-world network and its degree distribution is scale-free [44].

Also, protein networks, where proteins constitute the vertices and interactions between them (binding, catalysis or chemical modification etc.) defined as the edges of the networks, present scale-free degree distributions [30]. In this topology, few proteins have a high number of links (especially multifunctional ones), although most of them have few interactions. Interestingly, it is found that the highly connected (or interacting) proteins in 43 organisms are identical while the rest of the proteins are species specific [30]. The error-tolerance ability of scale-free architecture can enable such an evolution that preserves highly interacting proteins through time. Failures of highly connected nodes cannot be tolerated because of their deleterious effect to a large number of processes, causing the cell or organism to die. On the other hand, random failures of proteins having few links can be tolerated easily, and they can be seen as means of adaptability and flexibility. In addition, metabolic networks are shown to have a structure whereby there are a small number of connections between the clusters of highly connected nodes [47]. This topology gives a metabolic network modularity and additional robustness. Modularity refers to the fact that there are a few links between the

highly clustered proteins and these clusters can be seen as separate modules of cellular processes. This is also validated by compartmentalization and modularity characteristics of control of many cellular networks [48]. A failure of a highly connected protein affects the multi-protein complex around it dominantly; other complexes or processes are not affected as much because of fewer links between other clusters or multi-protein complexes. Because of this, such connectivity gives metabolic networks an additional robustness.

Recently, a mechanism is introduced for power laws in complex systems, which is referred as highly optimized tolerance (HOT) [49]. HOT systems are robust to perturbations they were designed to handle, yet fragile to unexpected perturbations and design flaws. The protein network is a HOT system because of its scale-free distribution. This network is very robust to random failures of nodes (proteins) in the network unless those failed are not highly-connected crucial ones. As an example, RNA polymerase which is a part the cellular machinery to transcribe DNA into mRNA in the cell, binds to several activator and regulatory proteins, which makes it a highly connected and crucial node in the protein network [21]. Failure of this protein is a very rare event, but cellular protein network is very fragile to its failure, its result is deleterious. In contrast, failure of several proteins can easily be tolerated by the protein network in the cell by macro-level mechanisms such as increasing the concentration of the activator molecules for the failed protein to reproduce it or switching to an alternative cellular state which does not need that protein.

Being scale-free is not universal for small-world networks, but there could be some constraints for being completely scale-free. This type of distribution is called truncated power law since the data conforms to a scale-free distribution up to a sharp cutoff of neighbor number, then the distribution of data is either an exponential or Gaussian decay [50]. In the power-law regime, the number of neighbors of any node is not limited. For some network cases, there are constraints which lead to fewer nodes having the high degrees expected from a scale-free distribution. These networks present a truncated power law distribution and their degree distribution is given by

$$y \sim x^{-\gamma} \exp(-x/b) \tag{3.1.}$$

where b is the truncation.

These constraints could be the aging of the vertices: Some vertices with high degree stop receiving new links. Although they remain in the network, they will no

longer receive new nodes, thus limiting the preferential attachment of new nodes expected in a scale-free distribution [50]. The network of movie actors is an example of a small world with truncated power law distribution. In this case, famous movie actors eventually stop acting in movies hence stop receiving new links in time. Although they are still part of the network, they receive much smaller number of links than expected from them. Figure 3.2.4a shows how aging affects the degree distribution of the network.

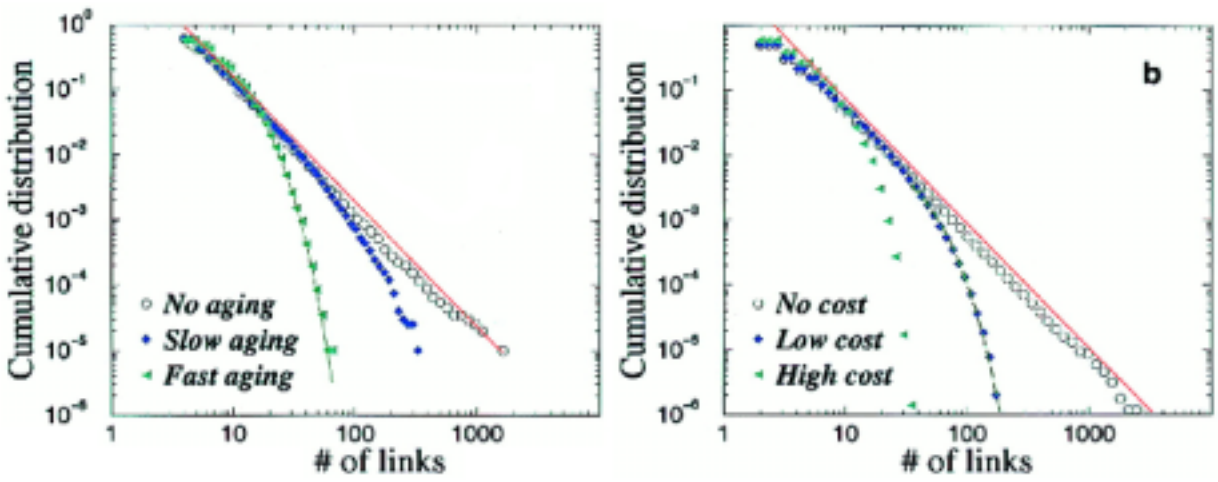


Figure 3.2.4. Physical constraints on P(k). (a) Aging of vertices (b) Cost of adding new links to highly connected nodes is also a constraint for scale-free distribution. This figure is adapted from Amaral et al. [50].

In the figure 3.2.4.a, circles show a scale-free distribution in which there is no limiting factor for preferential attachment of new nodes to nodes with high degree. With slow aging, nodes (denoted by blue squares) already having high degree stop receiving new nodes which is called the aging of vertices. In this case, the power law is truncated at a sharp cutoff. With fast aging, in the case denoted by green triangles, the distribution nearly became Gaussian which means that vertices stop receiving new nodes so early that they have no time to have high degrees.

Another constraint that limits the addition of new nodes to nodes of high degree is the increasing cost of adding new links to the vertices of high degree. Nodes having already high degree cannot receive new links because of reasons of efficiency [50]. This is exemplified very well in the network of world airports. There are some airports which are very busy and favored by all the airlines. However, because of space and time constraints, these airports have a limited number of links although they are hubs of the

network. Thus, they can no longer receive new links after a certain number of links. This limitation truncates the scale-free behavior of degree distribution of the network [50]. The effect of this constraint is exemplified in Figure 3.2.4.b. With no cost (denoted by circles), we see the scale-free distribution which means that there are no constraints limiting the neighbor number of nodes. With an intermediate cost denoted by blue squares, the distribution is a truncated power scale which means that new links can be allowed up to a cost value. With high costs, denoted by blue triangles, the distribution is Gaussian.

In a recent study, a new model is developed to explain the mechanism of truncated power law [51]. According to the model, systems having such distributions perform on sub-optimal levels (opposed to HOT systems performing at optimum), but they are more robust to any failure of nodes in the system (unlike HOT systems fragile to unforeseen failures). Such systems are called constrained optimization with limited deviations (COLD). COLD design is more tolerant than a HOT one by avoiding a total ruin (see example in section 3.2.2 for RNA polymerase), and accepting some loss in the average system performance [51].

The constraints to have truncated power law depend on the system or the network analyzed. So, different systems have different types of constraints. In the movie actor network, the aging of vertices is a constraint limiting the high degree nodes because of aging or death. But, world airports network have no such constraint but in this case high cost of adding new nodes is the constraint avoiding scale-free distribution. As we will see in the residue networks case, we will have different constraint for this system.

In this chapter, important features of networks and the patterns of these features in small-world networks are reviewed. To summarize, in order to decide whether a network is a small-world, we need to consider the two factors listed below;

i) Small-world networks have *C*s much larger than random networks, while *L*s do not vary much [26].

ii) Diameter of the network increases logarithmically with the number of vertices [45].

After these conditions are satisfied, the main problem is finding the degree distribution of the network. According to the distribution, it is necessary to explain the facts that push the network to have such a distribution and find the constraints.

Our specific interest in this project is the protein molecule as a network of its residues and we will do all the analysis listed above and explore what type of networks

proteins are. In the following chapter, one can find the methodology to convert proteins into networks, generate random networks and calculate the parameters of the resulting networks.

3.3 Network Model for Proteins

In order to treat a single protein as a network, we develop a method to convert a single protein into a network of its residues. We generate a random network which has the degree distribution (see section 3.2.3) with the original network but with a different connectivity. After generation these two networks using a single protein, we calculate particular parameters to characterize them. This work is done for all proteins in the dataset used for this project. In this chapter, all details of the methodologies are mentioned above.

3.3.1 Protein Network Generation

We convert a single protein into a network, vertices are the residues of the protein and edges are the interactions between them [25]. In order to link two residues in the protein, they should be located within a given cutoff distance. Such residues are assumed to be interacting and they are connected by an edge. In order to find all interacting residues, we place the primary sequence protein into both column and row of *N*x*N* matrix where *N* is the length of the protein. This matrix represents all possible pair of residues in the protein. The position of each amino acid is identified by that of its C_β atom. The distance between C_β atoms of residues in each pair are calculated and if their distance is smaller than a selected cutoff distance, they are assumed to be interacting with each other, in other words, they are in contact. Entry of the residue pair in the NxN matrix (contact map) will be 1 if the corresponding pair is in contact, otherwise, it will be 0. Hence, we have an adjacency matrix where each entry corresponds to a pair of residues and its value identifies whether the residues in this pair are in contact or not, depending on a selected cutoff distance. A number of networks for the same protein are

obtained whose connectivities are different according to the cutoff distance selected. All these networks generated with different cutoff distances are analyzed.

3.3.2 Random Network Generation

To interpret the network parameters calculated from protein networks, we need to generate their random counterparts. While generating a random network, we keep the number of neighbors of each residue the same, but change the neighbors of each residue. In order words, we rewire the network randomly such that;

i) We take a residue *n* whose degree is *k*, and we break all connections of this residue.

ii) For each of its neighbors, a random number is generated between 0 and protein length. Random numbers represent the residues indices of the residues in the primary sequence.

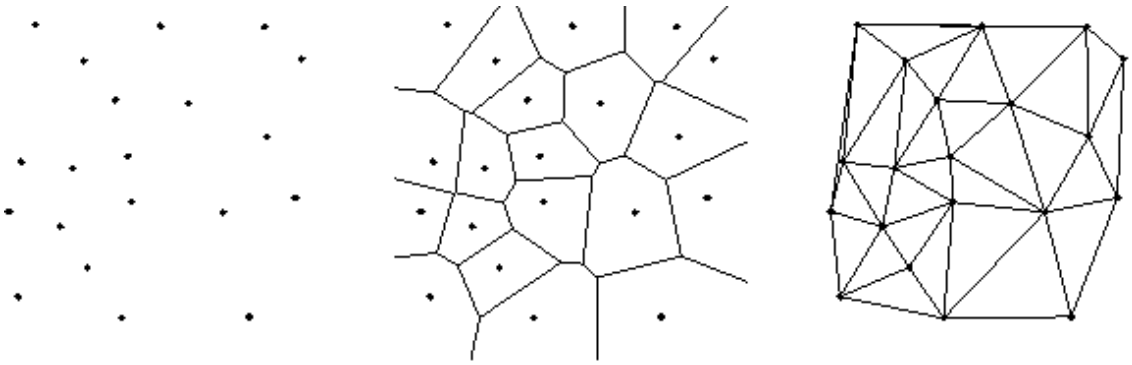
iii) The first randomly selected residue is taken and checked if it has enough neighbors for connection. These two residues are connected if the degree of the residue of random index is different from zero. On the other hand, if this residue is not available, another random number is generated for that connection.

iv) The number of neighbors of residue *n* and residue of random index is decreased by 1, since that connection is randomized. We apply the same procedure for all the neighbors of residue *n*. After completing the randomization of the connection of residue *n*, this whole procedure is repeated for all the residues in the protein.

There are some problems in this methodology while randomizing large proteins at high cutoff distances. As the number of nodes and the degree of the nodes increase, the number of possible networks having the same degree distribution but different connectivity decreases. Under such circumstances, the randomization is began with the residue with the highest degree and continue to the process with residues with lower degrees. The randomization process is also iterated until the algorithm finds another configuration desired.

3.3.3 Protein Network Generation Using DT

Another method is used to convert protein into a network of its residues. This method does not use any cutoff distance as a decision criterion for interaction, but it uses the geometrical placements of residues in three dimensional space. This method is called DT which is the dual of Voronoi. Given *N* points in a plane, Voronoi tessellation divides the domain in a set of polygonal regions, the boundaries of which are the perpendicular bisectors of the lines joining the points. Perpendicular bisector is the line that is perpendicular to the line connecting C_β atoms (C_α for Gly) of two residues and intersects it in the middle. Polygonal region of a node consists of points in the domain which are the closest points to this node than any other node in this domain. If one connects all the pairs of points for which the respective Voronoi regions share a common edge, one gets a DT. Also, circumcircle of each triangle does not contain any other node than the set of this triangle [52]. Figure 3.3.1 shows a number of nodes, its Voronoi tessellation, and the corresponding DT.



(a) (b) (c)

Figure 3.3.1. (a) A set of points in a plane is shown. (b) Voronoi tessellation of the set of points shown in (a). Each polyhedron of each node consists of the closest points to the corresponding node than any other node in the set. (c) The corresponding DT of the tessellation in (b). Note that vertices of each triangle are the nodes in the set and no triangle contains other nodes in the set except their vertices [53].

A set of nodes while using the DT is the C_β atoms (C_α for glycine) of each residue in a protein. DT of this set of points is generated. This is a triangulation in 3D, so there are tetrahedrons instead of triangles. The program named as "qhull" is used and it is available via anonymous ftp at ftp.ncsa.uiuc.edu [52].

3.3.4 Calculation of *L*

The shortest path length between any two residues of a protein is not directly deducible from the topology of the network. To calculate this, the powers of the contact map of the protein is used: if the shortest path between two nodes, *i* and *j*, is *d*-1, then ij^{th} entry in the d^{th} power of the adjacency matrix (contact map in our case) is equal to nonzero [54, 55]. Thus, the contact maps of the proteins are generated and the contact map of each protein is multiplied by itself until all of its entries are nonzero. Then, all the shortest path lengths are averaged out to obtain the *L* of the protein. The *L* of the network generated by DT is calculated in the same way. After triangulation, connectivity is converted into an adjacency matrix that is the contact map of the protein, and the same procedure is applied to calculate *L*.

This step is the bottleneck of this project since it is the procedure that takes the longest time especially for large proteins.

3.3.5 Calculation of *C*

As described in section 2.2., *C* of a network is determined by the average of clustering coefficient of every node in the network. The clustering coefficient (C_i) of i^{th} residue having *n* neighbors is given by

$$C_i = \frac{\text{Actual neighboring between neighbors of i}^{th}\text{ residue}}{n \times (n-1) \div 2} \tag{3.2.}$$

C of a residue network is calculated according to:

$$C = \frac{\sum_{i=0}^{N} C_i}{N} \tag{3.3}$$

The same methodology is valid for calculation of random residue networks and networks generated by DTs.

3.3.6 Degree Distribution

Degree of the i^{th} residue is obtained by counting the number of neighbors of that residue by including or excluding connectivity to see its effect on the degree distribution. The connectivity is excluded by disregarding its closest neighbors in the primary sequence (i-1 and i+1) of any residue.

3.3.7 Radial Distribution Function

Since the cutoff distance is central in our understanding of network properties of proteins, their radial distribution functions are also analyzed. Up to this point, the methodology followed to treat proteins as networks and how to calculate specific network parameters from these residue networks is summarized

The radial distribution function describes fluctuations in density around a given atom [40]. It is the average number of atoms found at a given distance in all directions. To calculate the radial distribution function, the procedure below is performed;

- The C_β atom of a residue in the protein is selected. A series of concentric spheres, each of them are set at a small fixed distance (Δr) apart are drawn around the selected atom
- The number of atoms inside each shell is counted and stored
- The number of atoms in each shell is divided by the volume of each shell ($4\pi r^2 \Delta r$).
- Procedure is repeated and averaged for all C_β atoms.

Radial distribution function can be deduced experimentally from X-ray or neutron diffraction studies. It is denoted as G(r) and calculated according to the formula given below;

$$G(r) = \sum_{r_s}^{R} \sum_{i=1}^{n} \frac{S_i(r)}{4\pi r^2 \Delta r} \tag{3.4}$$

where r denotes radius of the spheres drawn at each counting step, n is the protein length, r_s is the radius of the smallest sphere drawn, R is the radius of the largest sphere drawn, $S_i(r)$ is the total number of C_β atoms found around every C_β atoms of the protein in a shell of thickness Δr and distant to the i^{th} residue between r and $r + \Delta r$.

Radial distribution function is used to correlate the network parameters calculated for different number of networks with the favorable distances at which atoms reside relative to each other (coordination shells). Peaks in the plot of radial distribution function versus distance correspond to the place of coordination shells. The first peak corresponds to first coordination shell which is the most favorable distance between C_β atoms of residues, likewise for the second and third peaks. In the results and discussion section, the places of these coordination shells and their correlation with our other results will be mentioned in detail.

3.4 RESULTS AND DISCUSSION

In this study, 196 proteins are used whose sequence homology is less than 25%. This protein set was used earlier to predict contact maps of the proteins by Casadio et al. [5]. These proteins are selected from protein database PDB by a PDB-select algorithm [56]. Proteins with their chains used are listed in Table B in the Appendix.

3.4.1 Radial Distribution Function

As mentioned in section 3.6, radial distribution functions of proteins for C_β atoms of residues are found. The radial distribution function data coming from 196 proteins is combined to obtain an average function for our dataset. The smallest sphere is 0.5 Å, the largest sphere used is 50 Å in the calculation. The first and second coordination shells specific for C_β atoms of residues (C_α for glycine) are determined. Figure 3.4.1 shows a plot of the normalized radial distribution function $G(r)$ averaged out for the dataset. Normalization is achieved by dividing the number of atoms in each shell by the total area under the curve.

The first peak in $G(r)$, with maximum 5.5 Å and extending to 6.9 Å, corresponds to the first coordination shell. The second peak, with maximum 7.3 Å and extending to 8.6 Å, corresponds to second coordination shell and the third region with multiple maxima (9.9 Å and 10.9 Å) and extending to 11.8 Å is the third coordination shell. A cutoff value of ca. 7 Å is used in many studies where coarse graining of the proteins are

utilized. This value corresponds to the first coordination shell of the protein (6.8 Å for the set utilized here); i.e. the range within which residue pairs are found with the highest probability. A great portion of the contribution to this peak is due to chain connectivity; all (i, i+1) and most (i, i+2) pairs fall within this range. Non-bonded residue pairs also exist in this coordination shell. However, the contribution of non-bonded pairs to higher order coordination shells, which is usually neglected in studies employing a cutoff distance in proteins, may also be significant (e.g. in the face-centered cubic lattice structure, for which collisions can occur between third neighbors at intermediate densities in the vicinity of phase transition, there are six neighbors in the second shell and 24 neighbors in the third shell [57].

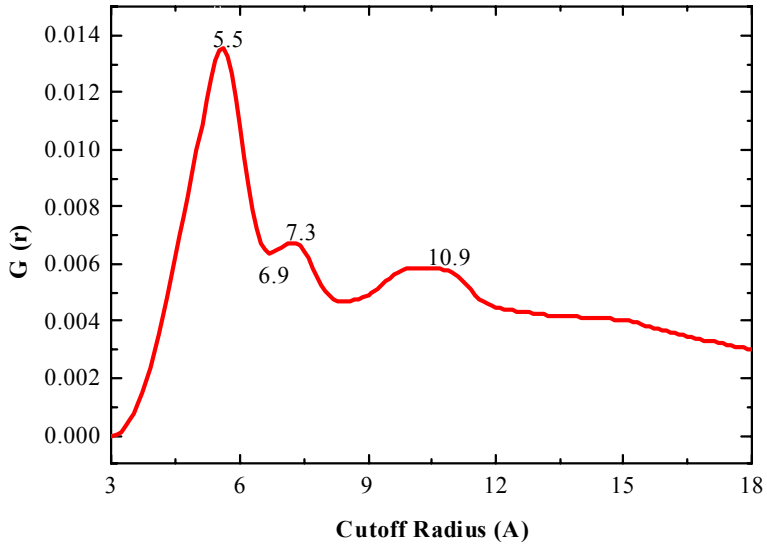


Figure 3.4.1. Radial Distribution Function of C_β atoms

3.4.2 Scaling of *L*

One of the determining characteristics of small-world networks is the scaling of the L with the logarithm of the size of the network. Since, residue networks are studied here, the scaling of L of proteins with their sizes is investigated. There are different residue networks at different cutoff distances, so the scaling analysis for a number of networks generated at different cutoff distances and by using DT is made. Figure 3.4.2 shows the scaling of the L of 196 proteins with the size of the proteins. This data is very

scattered and no information may be derived safely. To simplify the data, the proteins are grouped according to their sizes such that proteins with length M±20 are in the same group, with M being a multiple of 20. The average Ls of the proteins in one group are taken. Figure 3.4.3 shows this plot for four different cutoff distances and a curve fit is made to each. The general equation of the curves is

$$L = a \times \log(N) + b \tag{3.5}$$

Parameters of the above curve and goodness of fit (r^2) values of the curves for different cutoff distances are shown in Table 3.2.

Cutoff distance	a	b	r^2
5 Å	7.1 ± 0.7	-5.9 ± 1.6	0.8
7 Å	3.7 ± 0.3	-2.8± 0.6	0.9
9 Å	3.0 ± 0.2	-2.8 ± 0.4	0.9
12 Å	2.0 ± 0.1	-1.8 ± 0.3	0.9

Table 3.2. Parameters of L vs log(N) plot in Figure 4.3.

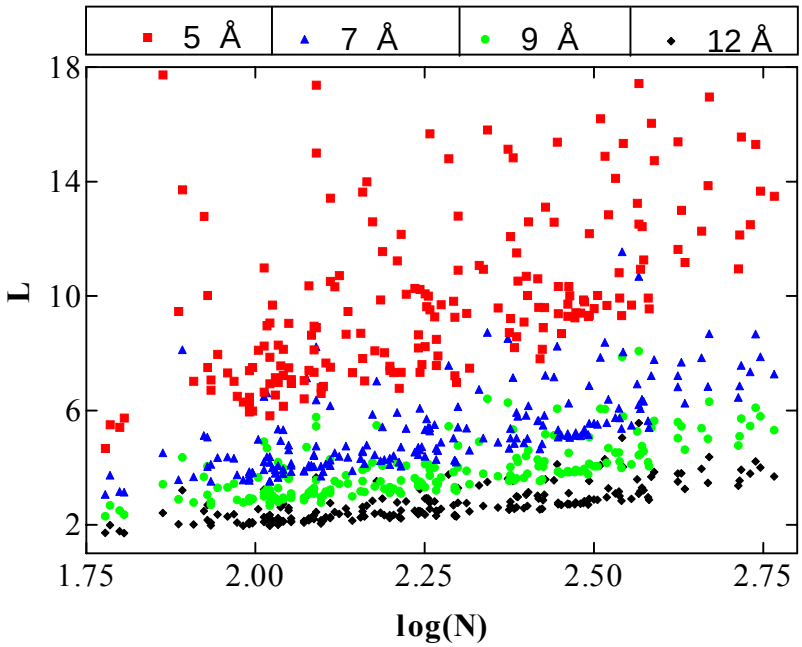


Figure 3.4.2. L versus protein length. Data is very scattered, no information can be derived. Different cutoff radii display with different symbols denoted above.

As can be seen in figure 3.4.3, network diameter changes with the logarithm of the size of the network. Hence, adding more nodes to the network does not affect the mean of the shortest distance traveled to go from one node to another significantly. The

network is organized such that new nodes make connections that connect them to the existing nodes with high degrees. By this organization, newly coming nodes easily adapt to the network structure and can transmit information easily by using their high degree neighbors. In proteins, such an organization can also be seen. This might be the result of the regular structure of the secondary structure elements of a protein and the long-range interactions which bridge these elements within the protein.

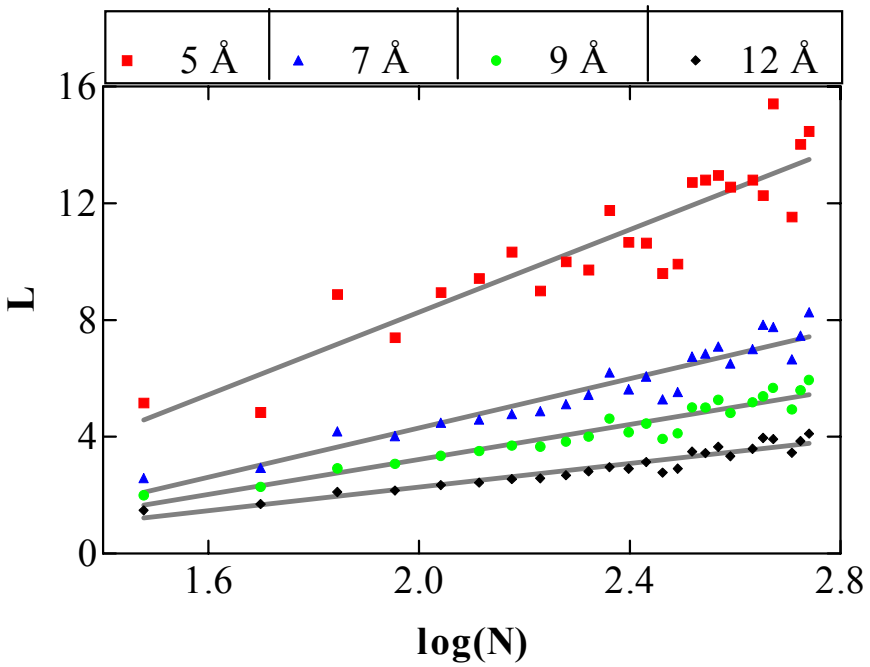


Figure 3.4.3. Scaling of L with protein length. Parameters of linear fit to the data is shown in table 4.2.

Also, networks using DTs from proteins are generated and the scaling of the L with the size of these networks is investigated. The L of all proteins in the dataset are calculated and plotted with respect to protein size. Again, since the data is very scattered, we group our data as described above. Figure 3.4.4 shows the scaling of L with protein size. The formula of the best-fit curve is

$$L = 1.869 \times \log N - 1.346 \qquad (3.6)$$

and goodness of fit (r^2) is 0.968.

These results indicate that residue networks have Ls which are scaled with the logarithm of the size of the proteins. One argument for this scaling can be as follows: interaction of residues determine the connectivity of the residue network, as the protein length increases, interaction of residues are not decreasing, there are still connections

(or interactions) that mediate the shorter path lengths. Such a behavior could be the result of the globular or high packed structure [58] of proteins, because loosely-packed structures cannot provide short path lengths as the network size increases.

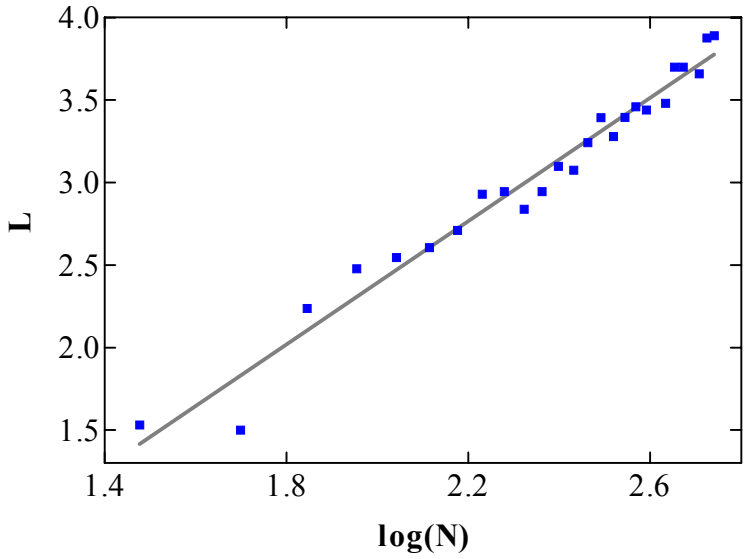


Figure 3.4.4. Scaling of *L* versus *N* in network generated by DT

3.4.3 *L* in Actual and Random Networks

The *L*s of small-world networks and of their randomly rewired counterparts do not differ much while the *C* differs significantly. In this section, we investigate how *L* differs between residue networks and their random counterparts. For this analysis, first each of 196 proteins is converted into networks using different cutoff distances. Then, the *L*s of these networks are calculated and averaged over all residue networks to calculate *L* at each cutoff distance. The mean of *L* of proteins at each cutoff distance versus cutoff distance is plotted. This is shown in Figure 3.4.5.and labeled with blue squares. The counterpart of this data from the randomized networks is also shown with circles.

We also calculate *L* of residue networks generated using DT. This is shown by a solid black line spanning the mean and the standard deviation of the edge distances. We calculate distance of each edge in all residue networks generated by DT. The latter is the average of the edge distances of the triangulated which is 9.2 + 1.8.

Thus it is observed that, one important property of small-world networks i.e. that *L* is on the same order of magnitude as random networks. However, the clustering coefficient also has to be analyzed to make a conclusive statement on the type of the network.

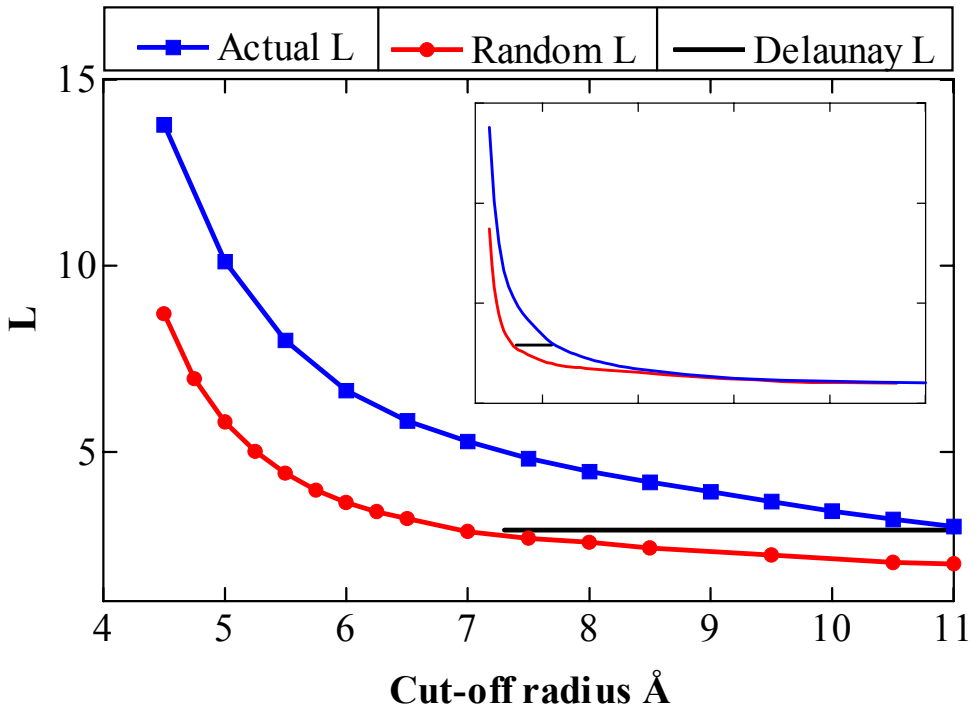


Figure 3.4.5. *L* in actual and random networks. Plots for all cut-ff radii is shown in the inset figure.

3.4.4 Clustering Coefficient in Actual and Random Networks

Clustering coefficient is a measure of neighbourhoodness of neighbors of any node in a network. In small-world networks, clustering coefficient is high because if node A is linked to node B and B is linked to node C, there is an increased probability that A will also be linked to C [59]. On the other hand, random networks are loosely clustered; there is no bias in the connectivity of the network as all the connections are random. To find how residue networks clustered, we calculate the clustering coefficient of 196 residue networks generated at different cutoff radii. Figure 3.4.6 displays average *C* at different cutoff distances (blue squares). The clustering coefficient of randomly rewired networks is calculated at different cutoffs (red circles).

The *C* of actual networks increases faster than of their random counterparts with increasing cutoff distance. There is one order of magnitude difference between the clustering densities of actual networks and their randomly rewired counterparts. The *C* of actual residue networks gives a very good fit to a five degree polynomial (R^2=0.96) whose inflection point is at 6.7 Å As the cutoff radius increases, the number of neighboring residues increases expectedly. Below 6.7 Å, new neighbors that are joining as the cutoff distance becomes larger increase the *C* significantly. However, newly joining neighbors do not affect the *C* of network as much. Moreover, the inflection point of this polynomial curve is where the first coordination shell ends (figure 3.4.1). This gives us an important clue about the organization of residues in proteins: Main clustering around a residue occurs between its first coordination shell neighbors and interactions within a cluster are mostly functionally and structurally important ones since they are between the neighbors occurring in the first coordination shell.

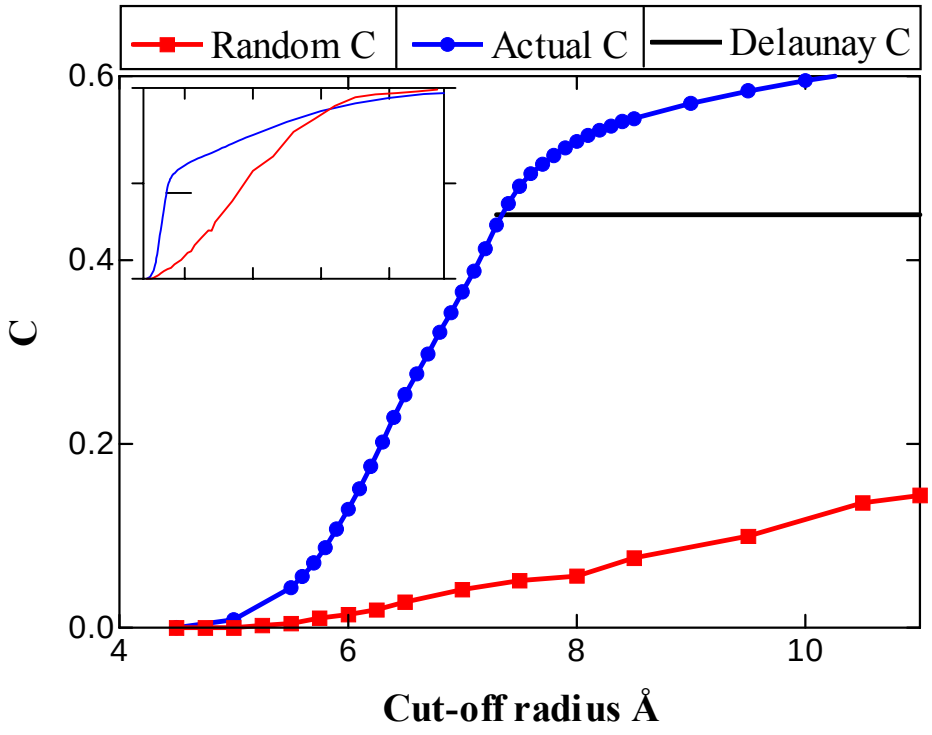


Figure 3.4.6. *C* in actual and random networks. Plots for all cutoff radii are shown in the inset figure.

In addition, most of the interactions present in these clusters are those within the secondary structure elements making them important factors that affect the formation of clusters.

Random networks lack such clusters which makes them weak to attacks to any node in the network. Networks having large clusters and random links (shortcuts) between these clusters respond to attacks much more strongly since losing one or more links within a cluster is more tolerable [46]. Having high C gives networks error tolerance ability. Since secondary structure elements are mostly responsible for the clustering within a protein, together with their stabilizing and functional roles of them, they also can help the protein to tolerate attacks to single residues occurring in the crowded environment of the cell; e.g. random collisions between atoms.

The logarithmic scaling of L with protein size in actual protein networks is observed in Section 3.4.2. Also, the differences between the Ls and clustering coefficients of actual protein networks and their randomly rewired counterparts are shown in section 3.4.3 and 3.4.4. The Ls of actual and random networks do not differ much, although their clustering coefficients differ significantly. Therefore, it is concluded that residue interaction based protein networks show small-world network behavior. One arrives at the same conclusion by comparing the Delaunay Triangulated proteins with their random counterparts.

It is worth noting the fact that the transition region in the C curve (Figure 3.4.6) ends around 9 Å, and that the correlation length obtained from a single exponential fit to the L curve (Figure 3.4.5) is 11.8 Å shows that the second and third coordination shells are important for local and global interactions, respectively (Figure 3.4.1).

3.4.5 Degree Distribution

The degree distributions of small-world networks are different from those of regularly and randomly organized networks (as mentioned in section 3.2.3). Next, the degree distributions of residue networks are determined.

Instead of presenting degree distributions of each residue network separately, a different methodology is applied to represent the degree distribution over all proteins in one graph. For each cutoff-distance that residue networks are generated, residues having k neighbors in each protein are counted and normalized by dividing by the length of the protein. Then, the degree distribution values are calculated according to,

$$\widetilde{P}(k) = \sum_{j=1}^{196} \widetilde{P}_j(k) \tag{3.7}$$

for each k where . $\widetilde{P}_j(k)$ is the normalized value.

Log-log plot of degree distribution of residue networks generated at 7 Å is shown in Figure 3.4.7.

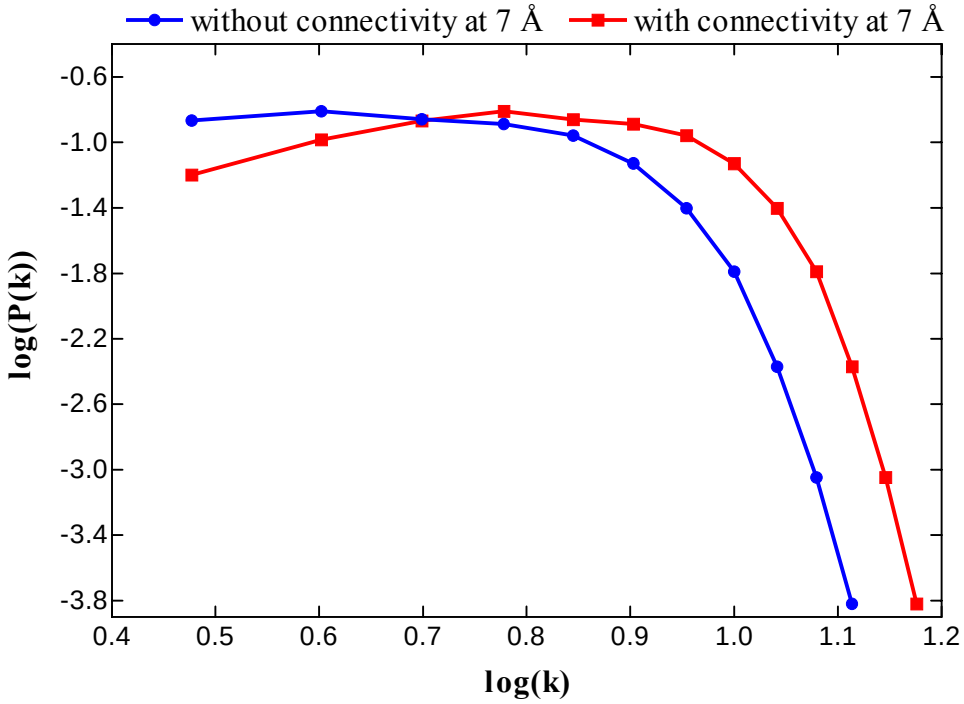


Figure 3.4.7. Average degree distribution of residue networks generated at 7 Å with and without connectivity.

In the figure, data denoted by blue circles include the primary sequence neighboring which is the connectivity information and the data denoted by red squares exclude the connectivity. Data with connectivity have a lower tail than that of without connectivity, since connectivity puts extra neighbors to every residue; residues having a low number of neighbors decrease and the average neighbor number increases. For both curves, it is observed that there is a fast decay of neighbor number having high degrees. This is a sign of truncation of power law because of physical constraints such as the excluded volume which put limits on the number of neighboring of residues.

7 Å is chosen to present the degree distribution of residue networks, because this value encompasses the first coordination shell in which functionally and structurally important interactions are most likely to occur. Since residue networks are built on the interaction of residues, it is convenient to examine the distribution at this cutoff radius.

Moreover, most studies which utilize a coarse-grained approach to the treatment of proteins were performed at this cutoff radius [57].

To understand how degree distributions differ between networks generated at different cutoff radii, three degree distributions are shown in Figure 3.4.8. Cutoff distances are chosen according to the peaks of radial distribution function; 5.5 Å is the peak of the first coordination shell, 6.9 Å is the end of first coordination shell and 11.8 is the end of third coordination shell. All the curves in figure 3.4.8 include the connectivity information.

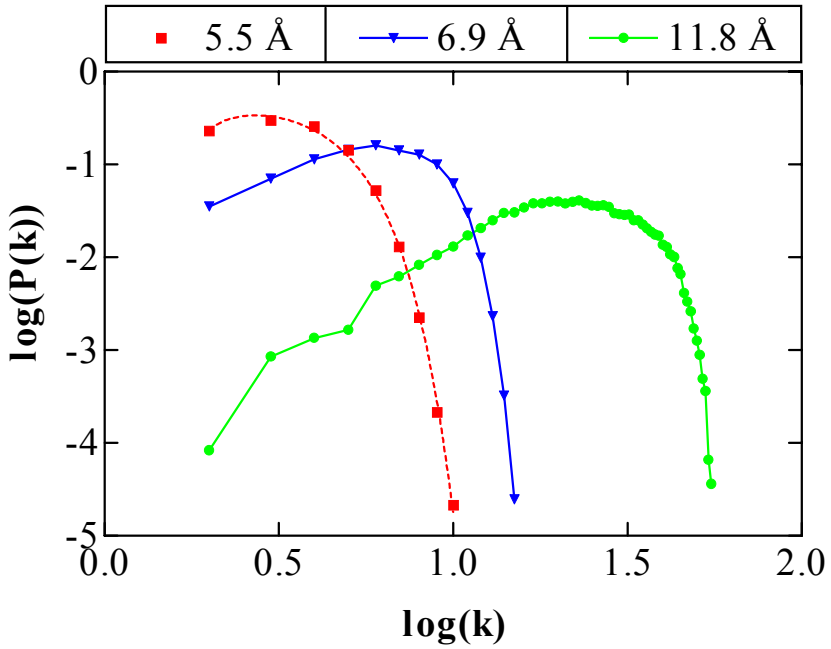


Figure 3.4.8. Log-log plot of degree distribution at three different cutoff radii.

The distribution is a truncated power law distribution at 5.5 Å, red dashed line is a curve fit of truncated power law equation shown in section 3.2.3 with an r^2 of 0.99. At other cutoff radii, although the curves have a lower tail at small neighbor numbers, there is a truncation as the neighbor number increases. This lowering of the tail is due to the increased effect of connectivity at large cutoff radii. In all distributions, there is a sharp cutoff at which the truncation begins as in the case of e.g. scientific collaboration networks [60].

The constraints preventing addition of more nodes to highly connected nodes after a certain neighbor number in residue networks give important clues for the structural organization of these networks. One constraint could be the excluded volume

effect. The number of residues that could be in the hypothetical sphere of a given radius of a residue is not unlimited. This is also valid for residue networks generated by DT since this networks shows the average edge distance which corresponds to a hypothetical sphere of radius of 9.2 Å drawn around each residue of the protein of interest. Polarity of residues also affects neighboring. Some residues cannot come close to each other because of the different polarity although their size allows the neighboring. The regularity of secondary structures of folded proteins might constitute another constraint for addition of new nodes to already highly connected nodes. A regular structure, like alpha helix or beta sheet, has a range of number of neighbors, and they cannot have more or less neighbors at given distances because of the regularity of the structure.

As discussed in section 3.2.3, small-world networks may have organized in two ways; small-world networks which is performing at optimal levels and error-tolerant unless these errors not attacking the crucial nodes are HOT systems, and small-world networks which are performing at sub-optimal levels but more tolerant to all kinds of attacks are called COLD systems. HOT systems have a degree distribution perfectly conforming to a power law whereas degree distribution of COLD systems conforms to truncated power laws. Residue networks can be classified as COLD systems because of their truncated power-law distribution. The strength of COLD design comes from the fact that there are less highly crucial nodes hence decreasing the probability of incoming attacks to these nodes. Since residue networks have these characteristics, they should also be more tolerant than HOT systems such as protein networks [30]. The constraints which could be limiting factors for neighbor number of certain residues are mentioned before. Excluded volume effect is an intrinsic limitation for the system; for any types of design, this constraint is always present. So, this effect can be seen as a cause of a COLD design in proteins or any type of macromolecule in general. The secondary structure elements which can also be classified as constraints are important for design. They control the number of neighbors of residues, creating a more stable and regular local environment. Stability is an important criterion for the protein to preserve its folded state and also, these elements are the means by which many functional processes performed by proteins such as binding. Thus, even in the absence of excluded volume effect, residue networks could not be a HOT system since such a configuration would not allow a stable and functional environment for the residues.

4. CONCLUSIONS

This thesis is aimed at predicting and understanding the protein structure using contact maps of proteins. Results show that residue contacts can give important information about the structure of proteins. The detailed conclusions of the parts of this thesis are mentioned in the following sections.

4.1 NN Predictor for Contacting Residues

Contacting residues in a protein are predicted using NNs. A multilayer perceptron with backpropagation algorithm is used for finding a correlation between the input and output of the network, which is used as a predictor if there is any. The inputs of the predictor are the selected physical and chemical features of residues along with or without of the selected features of their neighbors, the sequence separation along the chain and the length of the protein. The output of the predictor is the contact propensity of the residue pair input. In the previous studies, protein-based information was mainly used to predict contacting residues; residue-based information was not used much. Casadio et al. used evolutionary data and average hydrophobicity of residues to predict contact maps of proteins and they are six times better than a random predictor. Conversely, we encode size, charge and hydrophobicity information of residues to predict the contact propensity of residue pairs in a given content and separation along the primary chain. Our results show that this encoding is more than seven times better than a random predictor. Since the performance of the predictor developed here is better than the previous studies, it can be concluded that residue based information is relatively more correlated to contact propensity of residues of interest than other protein-based encodings used before.

Two identical networks are fed with different information and their performance are compared. One network (N1) is fed by individual size, charge and hydrophobicity of the residue pair of interest, while other network (N2) is fed by size, charge and average hydrophobicity over a residue window of seven neighbors. The results show that the network fed by individual hydrophobicity of residues is better than that fed by average hydrophobicity. Taking the average hydrophobicity of primary sequence neighbors might represent the local environment of the residues, and is expected to improve network accuracy since the local environment of the residues is very important for contact decision. However, it is observed from the results that averaging the hydrophobicities is not a proper way to encode the local environment of residues. Although hydrophobicity is a scalar quantity, it is the degree of non-polarity of a residue and it changes along the residue atoms. Also, the relative positions of the residues determine their non-polarity effects on each other. Two residues that are near along the primary chain but directed at different sides of the chain do not feel each other, although the averaging method assumes they do. Therefore, individual hydrophobicity information is more correlated with the contact propensity, but has the disadvantage of not expressing the effect of the environment.

A new method is developed to encode the local environment of residues. In this encoding, selected features of the three neighbors on each side of the residue of interest are used to represent the local environment. This network shows the best performance; it is seven times better than a random predictor. Based on the better performance of this representation, it can be said that introducing the size, charge and hydrophobicity information of neighboring residues as separate input nodes rather than averaging the selected features is a more appropriate strategy. Also, it is safer than averaging, because no information is lost if sufficient number of hidden nodes is used to learn the neighboring information. Since each feature of the neighbors is set to an input node, they are more successful in representing the effect of the neighbors to the residue of interest.

Our purpose in this work is not to develop a contact map predictor for practical purposes. Rather, we attempt to understand the factors influencing the contact decision of two residues in a given protein. It is found that encoding of physical and chemical features of residues and those of their neighbors improves the prediction. Therefore, this new encoding gives an insight on the factors affecting contact decision in the folding process.

Although, our results are compared with the result of Casadio and Fariselli [5], this comparison is not completely accurate since our contact definition is different than theirs. Here, there are approximately 98.4 times more non-contacts than contacts while, this ratio in their study is 60. This explains why our accuracy (<A>) values are lower than their accuracy values. However, since the improvement over a random prediction is less dependent on these ratios, these values are used for comparison.

N1 which has eight input nodes and encodes the size, charge, and individual hydrophobicity of the residue pair of interest has a lower prediction capability than N4 which encodes the local environment of residue pairs. This poor performance is due to the degeneracy of the training data, because the same residue pair with the same global properties can be both in contact or non-contact. This fact makes the learning process difficult for the network. Even architecture with enough complexity cannot achieve a remarkable generalization capability over all datasets of the problem. Therefore, a combination of features is required to separate contacting and non-contacting pairs, to present extra information to the predictor as well as to differentiate and learn these two cases. Encoding of local environment by using the physical and chemical features of neighboring residues serves this purpose.

To summarize, although our attempts to predict contacting residues in proteins is too weak to use for fold or structure prediction, a better prediction is attained by using physical and chemical features of residues and their neighbors. It adds a new dimension to this area by using parameters which were not considered before. Our predictor can achieve better results with the combination of other methods to contribute to the folding and design problems of proteins.

4.2 Characterization of Residue Networks

In the second part of this thesis, a protein is converted into a network of its interacting residues and it is found that this network is neither regular nor randomly organized, but it is a small-world network. Small-world networks are advantageous over both their random and regular counterparts, since they have shorter path lengths and error-tolerance. So, residues in folded proteins are not randomly organized; rather, their

distributions in space achieve a number of smart interactions that conform to a small-world topology and mediate their stability and functionality.

To perform their functions, proteins often exhibit a significant degree of flexibility and dynamics, which may occur on a wide range of time scales from femtoseconds to seconds [61]. Flexible parts, such as loop regions and side chains, are often involved in mediating specific protein-protein and protein-DNA interactions by changing their conformations upon establishment of specific contacts. As an example, let's take calmodulin molecule, which is a Ca^{+2} binding protein, crucial for muscle contraction. It has been shown that this molecule undergoes a large conformational change on the nanosecond time scale. In this conformational change, its central α-helix unwinds and two of its Ca^{+2} binding domains reorient themselves to make the molecule accessible for binding to target molecules [62]. Such conformational changes occurring on very short time scales require concerted actions of atoms and fast communication between residues. The latter cannot be achieved via the primary sequence of the protein; rather, shortcuts between residues generated by the certain folds of the protein are needed. If residues were all regularly packed in proteins, they would not mediate such short communication pathways between residues since regular networks always have longer path lengths due to a lack of shortcuts (see figure 2.1.). Thus, proteins form a structure, which provides fast information relay using residue interactions that are not necessarily adjacent in the primary sequence, but are close in the tertiary structure. In other words, proteins can carry information between remotely located regions by using a very small number of residues. Also, information relay has to be optimized on femtoseconds to nanosecond time scales and this might explain why proteins evolve to have structures whose interaction network conforms to the small-world topology.

Another important feature of small-world networks is their tolerance to random failures. Small-world networks having scale-free (power law) degree distributions, which are also called HOT systems, are tolerant to failures of nodes having low degree but are fragile to error on nodes having high degree. In turn, they perform at optimal levels. Protein networks in cells are HOT systems, since they are required to perform at optimal levels, because of the complexity of the tasks they execute.

In contrast to protein networks, it is shown here that the degree distribution of residue networks conforms to truncated power law that makes them COLD systems. Proteins have modular structures with α helices, β sheets or loop regions. These structural elements are regular and each residue within one of these elements has a

distinct number of neighbors most of which are interconnected; moreover, long-range interactions tie remote regions together. These structures can be seen as clusters of the residue networks. Residues have approximately the same number of neighbors; if the environment is crowded and the size and polarity of residues are favorable, residues interact more. This effect could be seen in the circular representation of residue network of 3chy protein (figure 2.2). The interactions are clustered as patches, which could be seen as the interactions between the secondary structure elements of the protein. Thus, proteins are COLD systems, because of their modular structural requirements.

This structure should be advantageous to proteins since it has been preserved over evolutionary time. It should be more important to tolerate attacks to any residue for proteins, since they are COLD systems. However, there is a paradox here. Some proteins, especially the ones that are functioning by binding over a few residues are not tolerant to attacks to these nodes; once these fail, the protein will be non-functional. Hemoglobin is a very common example: Changing one specific residue (Glu → Val mutation) has drastic effect over the structure of the protein. On the other hand, proteins which function by using a larger region, such as their loops or alpha helices (e.g. DNA binding proteins) are more tolerant to residue substitutions. So, there may be some differences when one looks at the degree distribution of these two different types of proteins; it might be expected that proteins which perform their functions via a small number of residues might conform to a HOTter design while others resemble COLD systems. Even in the hemoglobin example, the fragility of the system to failure of one residue is not enough to make it HOT design, because such crucial nodes exist in COLD systems, but their numbers is lower than that of HOT systems. Mutational studies and high number of proteins with high homologies confirm that proteins do not have a high number of hot and dangerous spots. Mild mutations in the DNA sequence that code a protein do not result in a total loss of function. Accumulation of such changes might generate a protein which can bind to a different molecule or carry out a different function. Hence, evolutionary plasticity of proteins requires a COLD design of proteins.

REFERENCES

1. Doruker, P., I. Bahar, C. Baysal, and B. Erman, *Collective Deformations in Proteins Determined by a Mode Analysis of Molecular Dynamics Trajectories.* Polymer, 2002. **43**: p. 431-439.

2. Micheal, J.E., *Protein Structure Prediction: Principles and Approaches*. 1996, New York: Oxford University Press. 1-26.

3. Vendruscolo, M., E. Kussell, and E. Domany, *Recovery of Protein Structure from Contact Maps.* Structure Fold. Des., 1997. **2**: p. 295-306.

4. Thomas, D.J., G. Casari, and C. Sander, *The Prediction of Protein Contacts from Multiple Sequence Alignments.* Protein Eng., 1996. **9**: p. 941-948.

5. Fariselli, P. and R. Casadio, *A Neural Network Based Predictor of Residue Contacts in Proteins.* Protein Eng., 1999. **12**: p. 15-21.

6. Sander, C. and R. Schneider, *Database of Homology-Derived Protein Structures and the Structural Meaning of Sequence Alignment.* Proteins, 1991. **9**: p. 56-68.

7. Zaki, M.J., S. Jin, and C. Bystroff. *Mining Residue Contacts in Proteins Using Local Structure Predictions*. in *IEEE International symposium on Bioinformatics and Biomedical Engineering*. 2000. Washington D.C.

8. http://vv.carleton.ca/~neil/neural/neuron-a.html.

9. Haykin, S., *Neural Networks: A Comprehensive Foundation*. Second ed. 1999, Upper Saddle River, N.J.: Prentice Hall.

10. http://www-personal.usyd.edu.au/~desm/afc-ann.html.

11. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/cs11/article2.html.

12. Hagan, M.T., H.B. Demuth, and M. Beale, *Neural Network Design*. 1996, Boston: PWS Publishing Company.

13. Bishop, C.M., *Neural Networks in Pattern Recognition*. 1996, New York: Oxford University Press.

14. Petersen, T.N., C. Lundegaard, M. Nielsen, H. Bohr, S. Brunak, G.P. Gippert, and O. Lund, *Prediction of Protein Secondary Structure at 80% Accuracy.* Proteins, 2000. **41**: p. 17-20.

15. Master, T., *Practical Neural Network Recipes in C++*. 1993: Academic Press.

16. http://www.biochem.ucl.ac.uk/bsm/sidechains/.

17. Baysal, C. and A.R. Atilgan, *Coordination topology and stability for the native and binding conformers of chymotrypsin inhibitor 2.* Proteins, 2001. **45**: p. 62-70.

18. Lee, B. and F. Richards, *The interpretation of protein structures: estimation of static accessibility.* J. Mol. Biol., 1971. **55**: p. 379-400.

19. http://pref.etfos.hr/scacor/.

20. Rose, G.D., A.R. Geselowitz, G.J. Lesser, R.H. Lee, and M.H. Zehfus, *Hydrophobicity of Amino Acid Residues in Globular Proteins.* Science, 1985. **229**: p. 834-838.

21. Alberts, B., D. Bray, J. Lewis, M. Raff, K. Roberts, and J.D. Watson, *Molecular Biology of the Cell*. Third Edition ed. 1994, New York: Garland Publishing.

22. Wang, J., B.D. Sykes, and R.O. Ryan, *Structural Basis for the Conformational Adaptability of Apolipophorin III, a Helix-Bundle Exchangeable Apolipoprotein.* Proc. Natl. Acad. Sci. U S A, 2002. **99**(3): p. 1188-93.

23. Stryer, L., *Biochemistry*. Fourth Edition ed. 1996, New York: W. H. Freeman and Company.

24. Strogatz, S.H., *Exploring Complex Networks.* Nature, 2001. **410**: p. 268-276.

25. Yilmaz, L.S. and A.R. Atilgan, *Identifying the Adaptive Mechanism in Globular Proteins: Fluctuations in Densely Packed Regions Manipulate Flexible Parts.* J. Chem. Phys., 2000. **113**: p. 4454-4464.

26. Watts, D.J. and S.H. Strogats, *Collective Dynamics of 'Small-World' Networks.* Nature, 1998. **393**: p. 440-442.

27. Broder, A.e.a., *Graph structure in the web.* Comput. Netw., 2000. **33**: p. 309-320.

28. Faloutsos, M.F., P. Faloutsos, C., *On power-law relationships of the Internet topology.* Computer Communication Review, 1999. **29**: p. 251-262.

29. Achacoso, T.B. and W.S. Yamamoto, *AY's Neuroanatomy of C. elegans for Computation*. 1992, Boca Raton, FL: CRC Press.

30. Jeong, H., B. Tombor, R. Albert, Z.N. Oltval, and A.-L. Barabasi, *The Large-Scale Organization of Metabolic Networks.* Nature, 2000. **407**: p. 651-654.

31. Richards, F.M. and W.A. Lim, *An Analysis of Packing in the Protein Folding Problem.* Q. Rev. Biophys., 1993. **26**(4): p. 423-98.

32. Raghunathan, G. and R.L. Jernigan, *Ideal architecture of residue packing and its observation in protein structures.* Prot. Sci., 1997. **6**(10): p. 2072-83.

33. Soyer, A., J. Chomilier, J.-P. Mornon, R. Jullien, and J.-F. Sadoc, *Voronoi Tessellation Reveals the Condensed Matter Character of Folded Proteins.* Phys. Rev. Lett., 2000. **85**: p. 3532-3535.

34. Liang, J. and K.A. Dill, *Are proteins Well-Packed?* Biophys. J., 2001. **81**: p. 751-766.

35. http://vlado.fmf.uni-lj.si/pub/networks/pajek/default.htm.

36. Erdös, P. and A. Renyi, *On the Evolution of Random Graphs.* Publ. Math. Inst. Hung. Acad. Sci., 1960. **5**: p. 17-61.

37. Delaunay, B.N., *Sur La Sphere vide.* Izv. Akad. Nauk SSSR, 1934. **7**: p. 793-800.

38. Voronoi, G.F., *Nouvelles Applications des Parametres Continus a la Theorie des Formes Quadratiques.* J. Reine Angew. Math., 1907. **133**(97-178).

39. Latora, V. and M. Marciori, *Efficient Behavior of Small-World Networks.* Phys. Rev. Lett., 2001. **87**(19).

40. Watts, D.J., *Small Worlds: the Dynamics of Networks Between Order and Randomness*. 1999, Princeton, NJ, USA: Princeton University Press.

41. Overbeek, R.e.a., *WIT: Integrated System for High-Throughput Genome Sequence Analysis and Metabolic Reconstruction.* Nucleic Acids Res., 2000. **28**: p. 123 -125.

42. http://www.ssec.wisc.edu/~billh/gbrain0.html.

43. Barthelemy, M. and L.A.N. Amaral, *Small-World Networks:Evidence for a Crossover Picture.* Phys. Rev. Lett., 1999. **82**: p. 3180-3183.

44. http://www.physicsweb.org/article/world/14/7/9/1/pw1407091.

45. Barabasi, A.-L. and R. Albert, *Emergence of Scaling in Random Networks.* Science, 1999. **286**: p. 509-512.

46. Albert, R., H. Jeong, and A.-L. Barabasi, *Error and Attact Tolerance of Complex Networks.* Nature, 2000. **406**: p. 378-381.

47. Maslov, S. and K. Sneppen, *Specificity and Stability in Topology of Protein Networks.* Science, 2002. **296**: p. 910-913.

48. Hartwell, L.H., J.J. Hopfield, S. Leibler, and A.W. Murray, *From Molecular to Modular Cell Biology.* Nature, 1999. **402**(6761 Suppl): p. C47-52.

49. Carlson, J.M. and J. Doyle, *Highly Optimized Tolerance: Robustness and Design in Complex Systems.* Phys. Rev. Lett., 2000. **84**: p. 2529-2532.

50. Amaral, L.A.N., M. Scala, A. Barthelemy, and H.E. Stanley, *Classes of Small-World Networks.* Proc. Natl. Acad. Sci. U S A, 2000. **97**: p. 11149-11152.

51. Newman, M.E.J., M. Girvan, and J.D. Farmer, *Optimal Design, Robustness and Risk Aversion.* to be submitted, 2002.

52. Facello, M.A., *Implementation of a Randomized Algorithm for Delaunay and Regular Triangulation in the Three Dimensions.* Comput. Aided Geom. Des., 1995. **12**: p. 349-370.

53. http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node3.html.

54. A.R. Atilgan, T.H., I. Bahar, B. Erman, *Correlated fluctuations in polymer networks.* Computational and theoretical polymer science, 1998. **8**: p. 55-59.

55. Cvetkovic, D., Rowlinson, P., Simic, S., *Eigenspaces of Graphs*. 1997, Cambridge: Cambridge University Press.

56. Hobohm, U., M. Scharf, R. Schneider, and C. Sander, *Selection of representative protein data sets.* Protein Science : a Publication of the Protein Society, 1992. **1**(3): p. 409-17.

57. Miyazawa, S. and R.L. Jernigan, *Residue-Residue Potentials with a Favorable Contact Pair Term and an Favorable High Packing Density Term, for Simulation and Threading.* J. Mol. Biol., 1996. **256**: p. 623-644.

58. Richards, F.M., *Areas, Volumes, Packing, and Protein Structures.* Annu. Rev. Biophys. Bioeng., 1977. **6**: p. 151-176.

59. Davidsen, J., H. Ebel, and S. Bornholdt, *Emergence of a Small World from Local Interactions: Modeling Acquaintance Networks.* Phys. Rev. Lett., 2002. **88**.

60. Newman, M.E., *The Structure of Scientific Collaboration Network.* Proc. Natl. Acad. Sci. U S A, 2001. **98**: p. 404-409.

61. http://www.hfsp.org/pubs/Awards_articles/Prompers.htm.

62. Wriggers, W., E. Mehler, F. Pitici, H. Weinstein, and K. Schulten, *Structure and Dynamics of Calmodulin in Solution.* Biophys. J., 1998. **74**: p. 1622-1639.

APPENDIX

153l		1esc		1mhc	A	1reg	X	2bnh	
1abr	B	1etc		1mhl	C	1rfb	A	2bop	A
1ade	A	1exg		1mla		1rib	A	2bpa	2
1aep		1fbr		1mml		1rva	A	2bpa	1
1aps		1fnf		1mmo	G	1scm	C	2cas	
1arb		1ghr		1mmo	B	1scu	B	2cpl	
1bbt	3	1gln		1mol	A	1ses	A	2end	
1ber	A	1gpr		1msc		1smn	A	2gmf	A
1bip		1hbq		1mse	C	1srs	A	2gst	A
1bnd	A	1hce		1mut		1sva	1	2kau	B
1bpl	A	1hcn	B	1nal	1	1svc	P	2liv	
1bpl	B	1hge	A	1nar		1svp	A	2mev	1
1bri	C	1hjr	A	1nhk	L	1tam		2min	B
1bvp	1	1hng	A	1noy	A	1tbr	R	2nac	A
1bw4		1hrz	A	1omp		1tfs		2ncm	
1cau	A	1hsl	A	1pba		1thv		2olb	A
1cau	B	1htm	D	1pbn		1thx		2pii	
1cew	I	1huc	B	1pcr	H	1tii	D	2ple	A
1chd		1hul	A	1pdn	C	1tiv		2pol	A
1chk	A	1hvk	A	1pgs		1tlk		2rsl	B
1cks	B	1ice	B	1pi2		1tnr	A	2scp	A
1cmb	A	1ice	A	1pkm		1tpg		2sil	
1cns	A	1ilk		1pkp		1trr	A	2tgi	
1cnv		1irl		1pls		1ttb	A	2tmv	P
1col	A	1knb		1pne		1urn	A	2vil	
1com	B	1kny	A	1pnk	A	1vca	A	3cd4	
1cse	I	1kpb	A	1pnk	B	1vhr	A	3pga	1
1ctn		1kpt	A	1prc	M	1vin		3pte	
1cus		1l17		1prc	C	1vmo	A	3sic	I
1cyu		1lau	E	1prt	F	1was		3tgl	
1dlc		1len	A	1prt	D	1xaa		4gcr	
1dpb		1lfb		1prt	B	1xyz	A	4rhv	3
1dsb	A	1lis		1ptd		1ypt	B	5tim	A
1dup	A	1lki		1ptv	A	1ytb	A	6fab	L
1dyn	A	1lpe		1ptx		1yua		8ruc	I
1ecp	A	1lts	D	1pvc	2	1zaa	C	8tln	E
1ede		1lts	A	1pvc	1	2aak			
1edg		1lxa		1pyp		2abd			
1eri	A	1mal		1rbu		2acg			
1erw		1mda	H	1rcb		2blt	A		

Table A. Proteins in the LRN protein subset are shown by their PDB codes and chains.

LRN	_	1hcn	B	1pkm		1ypt	B	1aih	A	1gds		1occ	D	1who		1doi		4sbv	A	1fjm	A	1eft	
153l		1hge	A	1pkp		1ytb	A	1air		1gnd		1occ	B	1xel		3chy		1atl	A	1fnc		1phg	
1abr	B	1hjr	A	1pls		1yua		1ako		1got	B	1occ	C	1xer		193L		1gen		2por		1qpg	
1ade	A	1hng	A	1pne		1zaa	C	1akz		1gow	A	1occ	A	1xik	A	1rcp	A	1iae		1irk		2amg	
1aep		1hrz	A	1pnk	A	2aak		1alk	A	1gpl		1ofg	A	1xjo		2aza	A	2gsq		8abp		1kbp	A
1aps		1hsl	A	1pnk	B	2abd		1alo		1gtm	A	1otg	A	1xsm		1hmt		1sac	A	2dln		2dkb	
1arb		1htm	D	1prc	M	2acg		1amm		1gym		1oun	A	1xva	A	1htp		1cfb		2ctc		1csh	
1bbt	3	1huc	B	1prc	C	2blt	A	1anu		1hav	A	1pax		1yas	A	1slt	B	1dyr		1gca		4enl	
1ber	A	1hul	A	1prt	F	2bnh		1aoc	A	1hcp		1pbw	A	1zxq		1poc		1fc2	D	1sbp		1hqa	A
1bip		1hvk	A	1prt	D	2bop	A	1apy	A	1hfh		1ped	A	1zym	A	1snc		1fua		8atc	A	1nhp	
1bnd	A	1ice	B	1prt	B	2bpa	2	1asz	A	1ihf	B	1pmi		2abh		1gtq	A	2abk		2cmd		1gcb	
1bpl	A	1ice	A	1ptd		2bpa	1	1axn		1iml		1pms		2arc	A	1pbx	A	9pap		1hvd		1pii	
1bpl	B	1ilk		1ptv	A	2cas		1beo		1iol		1pot		2bbi		3sdh	A	1thj	A	1gsa		2hpd	A
1bri	C	1irl		1ptx		2cpl		1bhm	A	1irs	A	1ppr	M	2fcr		1ash		1vid		1tag		3grs	
1bvp	1	1knb		1pvc	2	2end		1bmf	D	1iso		1psc		2fha		1vsd		2ayh		2acq		1gph	1
1bw4		1kny	A	1pvc	1	2gmf	A	1bmf	A	1ivd		1pud		2hpe	A	2fal		1gpc		1tca		1sat	
1cau	A	1kpb	A	1pyp		2gst	A	1bp1		1jac	A	1pue	E	2myr		8atc	B	2brd		1tah	A	1dnp	A
1cau	B	1kpt	A	1rbu		2kau	B	1bpy	A	1jer		1qap	A	2pfl		2hbg		2ak3	A	1quk		2pgd	
1cew	I	1l17		1rcb		2liv		1bro	A	1jpc		1qba		2pld	A	2mta	C	1nfp		2pia		6taa	
1chd		1lau	E	1reg	X	2mev	1	1btv		1jsw	A	1rai	D	2tbd		1sra		1pya	B	1poy	1	1dpg	A
1chk	A	1len	A	1rfb	A	2min	B	1bur	T	1jud		1rcy		2tys	B	1jcv		3pgm		1qor	A	1cow	A
1cks	B	1lfb		1rib	A	2nac	A	1cdq		1jvr		1rga		4rhn		2gdm		1din		1hmy		1byb	
1cmb	A	1lis		1rva	A	2ncm		1cem		1kap	P	1rgs		**COF**		1afb	1	1dhr		1nif		1smd	
1cns	A	1lki		1scm	C	2olb	A	1cex		1kaz		1rie				1mls		1gdo	A	1arv		8cat	A
1cnv		1lpe		1scu	B	2pii		1ckm	A	1kit		1rmd		5rxn		1phr		1cyd	A	1atp	E	1dpe	
1col	A	1lts	D	1ses	A	2ple	A	1cof		1klo		1rvv	1	1aaf		1esl		1bmt	A	2dld	A	1mmo	D
1com	B	1lts	A	1smn	A	2pol	A	1cpo		1kob	A	1ryc		1dtx		1hlb		1mrj		1pnr	A	1crl	
1cse	I	1lxa		1srs	A	2rsl	B	1cpq		1kuh		1ryt		1cdr		1jap	A	1ctm		1kif	A	1clc	
1ctn		1mal		1sva	1	2scp	A	1crk	A	1kve	B	1sei	A	1cea	A	1vhh		1nba	A	1mbb		1aoz	A
1cus		1mda	H	1svc	P	2sil		1csn		1kxu		1sft	A	1hcn	A	1bcf	A	1plq		2omf		3pmg	A
1cyu		1mhc	A	1svp	A	2tgi		1cyw		1l68		1shc	A	1pcn		1cyx		1tys		1rpa		2kau	C
1dlc		1mhl	C	1tam		2tmv	P	1def		1lbd		1sme	A	1fim		1obp	A	1dea	A	1fkx		4aah	A
1dpb		1mla		1tbr	R	2vil		1dek	A	1lbi	A	1stm	A	1umu		1std		1eny		2mnr		1pox	A
1dsb	A	1mml		1tfs		3cd4		1dhp	A	1lbu		1tcm	A	1mhl		3dfr		3fru	A	1ece	A	1aor	A
1dup	A	1mmo	G	1thv		3pga	1	1div		1lck	A	1tdt	A	9rnt		5p21		1ndh		1vsg	A	1sly	
1dyn	A	1mmo	B	1thx		3pte		1dkz	A	1lcl		1tf4	A	2psp		1rcf		2dri		1pea		1gof	
1ecp	A	1mol	A	1tii	D	3sic	I	1dor	A	1lid		1tfe		1put		1mka	A	1dih		1cdo	A	1trk	A
1ede		1msc		1tiv		3tgl		1dos	A	1lit		1tgx	A	2fd2		1rci		1qrd	A	2btf	A	1cyg	
1edg		1mse	C	1tlk		4gcr		1drw		1lrv		1uae		1fkj		1prr		2hhm	A	1hpm		1lcf	
1eri	A	1mut		1tnr	A	4rhv	3	1dxy		1lzr		1uby		2cdv		1fcd	C	1daa	A	1buc	A	1oac	A
1erw		1nal	1	1tpg		5tim	A	1eal		1mbd		1ucw	A	1msa		2prd		2prk		1ubs	B	2tmd	A
1esc		1nar		1trr	A	6fab	L	1ebp	A	1mhy	G	1ulp		1rtp		1cid		1nip	A	1nsc	A	8acn	
1etc		1nhk	L	1ttb	A	8ruc	I	1ecr	A	1msf	C	1uxy		1ccr		1dlh	A	2ebn		1pbe		1gpb	
1exg		1noy	A	1urn	A	8tln	E	1edh	A	1mty	B	1vcc		2hmz	A	1lfa	A	1tml		4xia	A	1bgl	A
1fbr		1omp		1vca	A	**TS97**	_	1emk		1nfa		1vhi	A	1sri	B	2stv		1han		1svb			
1fnf		1pba		1vhr	A	1aa6		1etp	A	1nfk	A	1vls		1bp2		1xnb		1scu	A	1oyc			
1ghr		1pbn		1vin		1aa8	A	1eur		1nkl		1vnc		2mad	L	2sas		1amp		1inp			
1gln		1pcr	H	1vmo	A	1ad2		1fbt	A	1nox		1vok	A	4fgf		1gky		2cyp		1chm	A		
1gpr		1pdn	C	1was		1ad3	A	1fib		1npo	A	1vsc	A	7rsa		1dlh	B	2ora		1fcd	A		
1hbq		1pgs		1xaa		1afr	A	1fro	A	1nsy	A	1wba		2phy		1isc	A	1sch	A	1oxa			
1hce		1pi2		1xyz	A	1agr	E	1gal		1nzy	A	1whi		2ccy	A	1tup	B	1ctt		1psd	A		

Table B. Three protein subsets used in the first part are shown with their PDB codes and chains.

Validation Proteins											
Val Set 1			Val Set 2			Val Set 3			Val Set 4		
Protein Name	Protein Length	Contact Number	Protein Name	Protein Length	Contact Number	Protein Name	Protein Length	Contact Number	Protein Name	Protein Length	Contact Number
1tgx	60	82	1xer	102	176	1aih	170	174	1ecr	305	372
1psc	69	103	1rga	104	150	1wba	171	318	1dor	311	521
2bbi	71	77	1msf	105	73	2fha	172	154	1nfk	312	513
1hcp	75	77	2pld	105	104	2fcr	173	275	1ppr	312	300
1iml	76	86	1jpc	108	174	1amm	174	323	1ucw	315	475
1cdq	77	111	1jer	109	159	1aoc	175	255	1ckm	317	504
1kve	77	85	1irs	112	157	1fro	176	194	2abh	321	551
1vcc	77	108	4rhn	115	144	1nfa	178	207	1pot	322	492
1nkl	78	72	1rmd	116	126	1pbw	184	184	1axn	323	367
1npo	81	146	1hfh	120	209	1fbt	186	265	1bpy	326	404
1pue	88	85	2pfl	121	183	1etp	190	249	1sme	329	619
1ihf	94	70	1whi	122	229	1ryt	190	199	1dxy	332	478
1who	94	135	1bur	123	131	1vok	192	284	1xel	338	564
1beo	98	107	1otg	125	116	1zxq	192	351	1got	339	697
2hpe	99	140	1oun	125	152	1shc	195	261	1aa8	340	526
			1eal	127	170	1vsc	196	314	1uxy	340	588
			1rie	127	208	1bhm	198	292	1xik	340	364
			1agr	128	117	1cex	200	333	1afr	345	391
			1cpq	129	113	1nox	200	220	1uby	348	339
			1lzr	130	181	1ebp	211	330	1pax	350	500
			1sei	130	190	1edh	211	371	1ped	351	698
			1lid	131	173	1lbu	214	301	1air	352	707
			1lit	131	207	1dkz	215	286	1kob	352	509
			1kuh	132	193	1hav	216	375	1eur	361	724
			1jac	133	252	1emk	220	400	1cem	363	571
			2tbd	134	185	1jud	220	298	1dos	369	596
			1cof	135	188	1akz	223	311	1pud	372	595
			1pms	135	163	1ad2	224	310	1kaz	378	608
			1jvr	137	50	1occ	227	266	1crk	380	563
			1anu	138	246	1lrv	233	275	1ofg	381	570
			1vhi	139	175	1lbd	238	227	1sft	382	640
			1lcl	141	228	1dek	240	286	1mty	384	387
			1stm	141	231	1zym	247	323	1ivd	388	858
			1tfe	142	160	1fib	249	427	2tys	396	721
			1occ	144	58	1tdt	256	432	1iso	414	660
			1rai	145	204	1yas	256	414	1gtm	417	691
			1vls	146	102	1occ	261	207	1uae	418	843
			1def	147	267	1rgs	264	385	1gnd	430	652
			1div	149	172	1ako	268	407	1gpl	432	793
			1gds	151	126	1nzy	269	354	1pmi	440	751
			1rcy	151	275	1nsy	271	339	1ad3	446	655
			1ulp	152	269	1drw	272	411	1alk	449	860
			1mbd	153	125	1kxu	276	280	1bp1	456	699
			1rvv	154	233	1bro	277	433	1jsw	459	612
			1btv	159	212	1iol	284	354	1bmf	467	753
			1cyw	159	253	1xsm	288	321	1kap	470	849
			1xjo	160	233	1qap	289	415	1bmf	487	770
			1apy	161	191	1ryc	291	382	1gow	489	790
			2arc	161	238	1dhp	292	483	1asz	490	689
			1klo	162	303	1xva	292	391	1occ	514	633
			1l68	162	167	1csn	293	389	2myr	519	169
			1lck	164	236	1gym	296	485	1vnc	576	928
			1mhy	167	131	1lbi	296	488	1gal	581	1108
						1cpo	299	407	1tf4	605	1041
									1tcm	686	1291
									1aa6	696	1276
									1kit	757	1503
									1qba	863	1575
									1alo	908	1790

Table C. Proteins in the validation set TS97. Contact numbers are obtained using the contact definition in section 2.3.3 in the second part of the thesis and contacting residues whose sequence separation is less than four residues are not included.

REFERENCES

1. Doruker, P., I. Bahar, C. Baysal, and B. Erman, *Collective Deformations in Proteins Determined by a Mode Analysis of Molecular Dynamics Trajectories.* Polymer, 2002. **43**: p. 431-439.

2. Micheal, J.E., *Protein Structure Prediction: Principles and Approaches*. 1996, New York: Oxford University Press. 1-26.

3. Vendruscolo, M., E. Kussell, and E. Domany, *Recovery of Protein Structure from Contact Maps.* Structure Fold. Des., 1997. **2**: p. 295-306.

4. Thomas, D.J., G. Casari, and C. Sander, *The Prediction of Protein Contacts from Multiple Sequence Alignments.* Protein Eng., 1996. **9**: p. 941-948.

5. Fariselli, P. and R. Casadio, *A Neural Network Based Predictor of Residue Contacts in Proteins.* Protein Eng., 1999. **12**: p. 15-21.

6. Sander, C. and R. Schneider, *Database of Homology-Derived Protein Structures and the Structural Meaning of Sequence Alignment.* Proteins, 1991. **9**: p. 56-68.

7. Zaki, M.J., S. Jin, and C. Bystroff. *Mining Residue Contacts in Proteins Using Local Structure Predictions*. in *IEEE International symposium on Bioinformatics and Biomedical Engineering*. 2000. Washington D.C.

8. http://vv.carleton.ca/~neil/neural/neuron-a.html.

9. Haykin, S., *Neural Networks: A Comprehensive Foundation*. Second ed. 1999, Upper Saddle River, N.J.: Prentice Hall.

10. http://www-personal.usyd.edu.au/~desm/afc-ann.html.

11. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/cs11/article2.html.

12. Hagan, M.T., H.B. Demuth, and M. Beale, *Neural Network Design*. 1996, Boston: PWS Publishing Company.

13. Bishop, C.M., *Neural Networks in Pattern Recognition*. 1996, New York: Oxford University Press.

14. Petersen, T.N., C. Lundegaard, M. Nielsen, H. Bohr, S. Brunak, G.P. Gippert, and O. Lund, *Prediction of Protein Secondary Structure at 80% Accuracy.* Proteins, 2000. **41**: p. 17-20.

15. Master, T., *Practical Neural Network Recipes in C++*. 1993: Academic Press.

16. http://www.biochem.ucl.ac.uk/bsm/sidechains/.

17. Baysal, C. and A.R. Atilgan, *Coordination topology and stability for the native and binding conformers of chymotrypsin inhibitor 2.* Proteins, 2001. **45**: p. 62-70.

18. Lee, B. and F. Richards, *The interpretation of protein structures: estimation of static accessibility.* J. Mol. Biol., 1971. **55**: p. 379-400.

19. http://pref.etfos.hr/scacor/.

20. Rose, G.D., A.R. Geselowitz, G.J. Lesser, R.H. Lee, and M.H. Zehfus, *Hydrophobicity of Amino Acid Residues in Globular Proteins.* Science, 1985. **229**: p. 834-838.

21. Alberts, B., D. Bray, J. Lewis, M. Raff, K. Roberts, and J.D. Watson, *Molecular Biology of the Cell*. Third Edition ed. 1994, New York: Garland Publishing.

22. Wang, J., B.D. Sykes, and R.O. Ryan, *Structural Basis for the Conformational Adaptability of Apolipophorin III, a Helix-Bundle Exchangeable Apolipoprotein.* Proc. Natl. Acad. Sci. U S A, 2002. **99**(3): p. 1188-93.

23. Stryer, L., *Biochemistry*. Fourth Edition ed. 1996, New York: W. H. Freeman and Company.

24. Strogatz, S.H., *Exploring Complex Networks.* Nature, 2001. **410**: p. 268-276.

25. Yilmaz, L.S. and A.R. Atilgan, *Identifying the Adaptive Mechanism in Globular Proteins: Fluctuations in Densely Packed Regions Manipulate Flexible Parts.* J. Chem. Phys., 2000. **113**: p. 4454-4464.

26. Watts, D.J. and S.H. Strogats, *Collective Dynamics of 'Small-World' Networks.* Nature, 1998. **393**: p. 440-442.

27. Broder, A.e.a., *Graph structure in the web.* Comput. Netw., 2000. **33**: p. 309-320.

28. Faloutsos, M.F., P. Faloutsos, C., *On power-law relationships of the Internet topology.* Computer Communication Review, 1999. **29**: p. 251-262.

29. Achacoso, T.B. and W.S. Yamamoto, *AY's Neuroanatomy of C. elegans for Computation*. 1992, Boca Raton, FL: CRC Press.

30. Jeong, H., B. Tombor, R. Albert, Z.N. Oltval, and A.-L. Barabasi, *The Large-Scale Organization of Metabolic Networks.* Nature, 2000. **407**: p. 651-654.

31. Richards, F.M. and W.A. Lim, *An Analysis of Packing in the Protein Folding Problem.* Q. Rev. Biophys., 1993. **26**(4): p. 423-98.

32. Raghunathan, G. and R.L. Jernigan, *Ideal architecture of residue packing and its observation in protein structures.* Prot. Sci., 1997. **6**(10): p. 2072-83.

33. Soyer, A., J. Chomilier, J.-P. Mornon, R. Jullien, and J.-F. Sadoc, *Voronoi Tessellation Reveals the Condensed Matter Character of Folded Proteins.* Phys. Rev. Lett., 2000. **85**: p. 3532-3535.

34. Liang, J. and K.A. Dill, *Are proteins Well-Packed?* Biophys. J., 2001. **81**: p. 751-766.

35. http://vlado.fmf.uni-lj.si/pub/networks/pajek/default.htm.

36. Erdös, P. and A. Renyi, *On the Evolution of Random Graphs.* Publ. Math. Inst. Hung. Acad. Sci., 1960. **5**: p. 17-61.

37. Delaunay, B.N., *Sur La Sphere vide.* Izv. Akad. Nauk SSSR, 1934. **7**: p. 793-800.

38. Voronoi, G.F., *Nouvelles Applications des Parametres Continus a la Theorie des Formes Quadratiques.* J. Reine Angew. Math., 1907. **133**(97-178).

39. Latora, V. and M. Marciori, *Efficient Behavior of Small-World Networks.* Phys. Rev. Lett., 2001. **87**(19).

40. Watts, D.J., *Small Worlds: the Dynamics of Networks Between Order and Randomness*. 1999, Princeton, NJ, USA: Princeton University Press.

41. Overbeek, R.e.a., *WIT: Integrated System for High-Throughput Genome Sequence Analysis and Metabolic Reconstruction.* Nucleic Acids Res., 2000. **28**: p. 123 -125.

42. http://www.ssec.wisc.edu/~billh/gbrain0.html.

43. Barthelemy, M. and L.A.N. Amaral, *Small-World Networks:Evidence for a Crossover Picture.* Phys. Rev. Lett., 1999. **82**: p. 3180-3183.

44. http://www.physicsweb.org/article/world/14/7/9/1/pw1407091.

45. Barabasi, A.-L. and R. Albert, *Emergence of Scaling in Random Networks.* Science, 1999. **286**: p. 509-512.

46. Albert, R., H. Jeong, and A.-L. Barabasi, *Error and Attact Tolerance of Complex Networks.* Nature, 2000. **406**: p. 378-381.

47. Maslov, S. and K. Sneppen, *Specificity and Stability in Topology of Protein Networks.* Science, 2002. **296**: p. 910-913.

48. Hartwell, L.H., J.J. Hopfield, S. Leibler, and A.W. Murray, *From Molecular to Modular Cell Biology.* Nature, 1999. **402**(6761 Suppl): p. C47-52.

49. Carlson, J.M. and J. Doyle, *Highly Optimized Tolerance: Robustness and Design in Complex Systems.* Phys. Rev. Lett., 2000. **84**: p. 2529-2532.

50. Amaral, L.A.N., M. Scala, A. Barthelemy, and H.E. Stanley, *Classes of Small-World Networks.* Proc. Natl. Acad. Sci. U S A, 2000. **97**: p. 11149-11152.

51. Newman, M.E.J., M. Girvan, and J.D. Farmer, *Optimal Design, Robustness and Risk Aversion.* to be submitted, 2002.

52. Facello, M.A., *Implementation of a Randomized Algorithm for Delaunay and Regular Triangulation in the Three Dimensions.* Comput. Aided Geom. Des., 1995. **12**: p. 349-370.

53. http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node3.html.

54. A.R. Atilgan, T.H., I. Bahar, B. Erman, *Correlated fluctuations in polymer networks.* Computational and theoretical polymer science, 1998. **8**: p. 55-59.

55. Cvetkovic, D., Rowlinson, P., Simic, S., *Eigenspaces of Graphs*. 1997, Cambridge: Cambridge University Press.

56. Hobohm, U., M. Scharf, R. Schneider, and C. Sander, *Selection of representative protein data sets.* Protein Science : a Publication of the Protein Society, 1992. **1**(3): p. 409-17.

57. Miyazawa, S. and R.L. Jernigan, *Residue-Residue Potentials with a Favorable Contact Pair Term and an Favorable High Packing Density Term, for Simulation and Threading.* J. Mol. Biol., 1996. **256**: p. 623-644.

58. Richards, F.M., *Areas, Volumes, Packing, and Protein Structures.* Annu. Rev. Biophys. Bioeng., 1977. **6**: p. 151-176.

59. Davidsen, J., H. Ebel, and S. Bornholdt, *Emergence of a Small World from Local Interactions: Modeling Acquaintance Networks.* Phys. Rev. Lett., 2002. **88**.

60. Newman, M.E., *The Structure of Scientific Collaboration Network.* Proc. Natl. Acad. Sci. U S A, 2001. **98**: p. 404-409.

61. http://www.hfsp.org/pubs/Awards_articles/Prompers.htm.

62. Wriggers, W., E. Mehler, F. Pitici, H. Weinstein, and K. Schulten, *Structure and Dynamics of Calmodulin in Solution.* Biophys. J., 1998. **74**: p. 1622-1639.

APPENDIX

153l		1esc		1mhc	A	1reg	X	2bnh	
1abr	B	1etc		1mhl	C	1rfb	A	2bop	A
1ade	A	1exg		1mla		1rib	A	2bpa	2
1aep		1fbr		1mml		1rva	A	2bpa	1
1aps		1fnf		1mmo	G	1scm	C	2cas	
1arb		1ghr		1mmo	B	1scu	B	2cpl	
1bbt	3	1gln		1mol	A	1ses	A	2end	
1ber	A	1gpr		1msc		1smn	A	2gmf	A
1bip		1hbq		1mse	C	1srs	A	2gst	A
1bnd	A	1hce		1mut		1sva	1	2kau	B
1bpl	A	1hcn	B	1nal	1	1svc	P	2liv	
1bpl	B	1hge	A	1nar		1svp	A	2mev	1
1bri	C	1hjr	A	1nhk	L	1tam		2min	B
1bvp	1	1hng	A	1noy	A	1tbr	R	2nac	A
1bw4		1hrz	A	1omp		1tfs		2ncm	
1cau	A	1hsl	A	1pba		1thv		2olb	A
1cau	B	1htm	D	1pbn		1thx		2pii	
1cew	I	1huc	B	1pcr	H	1tii	D	2ple	A
1chd		1hul	A	1pdn	C	1tiv		2pol	A
1chk	A	1hvk	A	1pgs		1tlk		2rsl	B
1cks	B	1ice	B	1pi2		1tnr	A	2scp	A
1cmb	A	1ice	A	1pkm		1tpg		2sil	
1cns	A	1ilk		1pkp		1trr	A	2tgi	
1cnv		1irl		1pls		1ttb	A	2tmv	P
1col	A	1knb		1pne		1urn	A	2vil	
1com	B	1kny	A	1pnk	A	1vca	A	3cd4	
1cse	I	1kpb	A	1pnk	B	1vhr	A	3pga	1
1ctn		1kpt	A	1prc	M	1vin		3pte	
1cus		1l17		1prc	C	1vmo	A	3sic	I
1cyu		1lau	E	1prt	F	1was		3tgl	
1dlc		1len	A	1prt	D	1xaa		4gcr	
1dpb		1lfb		1prt	B	1xyz	A	4rhv	3
1dsb	A	1lis		1ptd		1ypt	B	5tim	A
1dup	A	1lki		1ptv	A	1ytb	A	6fab	L
1dyn	A	1lpe		1ptx		1yua		8ruc	I
1ecp	A	1lts	D	1pvc	2	1zaa	C	8tln	E
1ede		1lts	A	1pvc	1	2aak			
1edg		1lxa		1pyp		2abd			
1eri	A	1mal		1rbu		2acg			
1erw		1mda	H	1rcb		2blt	A		

Table A. Proteins in the LRN protein subset are shown by their PDB codes and chains.

LRN	_	1hcn	B	1pkm		1ypt	B	1aih	A	1gds		1occ	D	1who		1doi		4sbv	A	1fjm	A	1eft	
153l		1hge	A	1pkp		1ytb	A	1air		1gnd		1occ	B	1xel		3chy		1atl	A	1fnc		1phg	
1abr	B	1hjr	A	1pls		1yua		1ako		1got	B	1occ	C	1xer		193L		1gen		2por		1qpg	
1ade	A	1hng	A	1pne		1zaa	C	1akz		1gow	A	1occ	A	1xik	A	1rcp	A	1iae		1irk		2amg	
1aep		1hrz	A	1pnk	A	2aak		1alk	A	1gpl		1ofg	A	1xjo		2aza	A	2gsq		8abp		1kbp	A
1aps		1hsl	A	1pnk	B	2abd		1alo		1gtm	A	1otg	A	1xsm		1hmt		1sac	A	2dln		2dkb	
1arb		1htm	D	1prc	M	2acg		1amm		1gym		1oun	A	1xva	A	1htp		1cfb		2ctc		1csh	
1bbt	3	1huc	B	1prc	C	2blt	A	1anu		1hav	A	1pax		1yas	A	1slt	B	1dyr		1gca		4enl	
1ber	A	1hul	A	1prt	F	2bnh		1aoc	A	1hcp		1pbw	A	1zxq		1poc		1fc2	D	1sbp		1hqa	A
1bip		1hvk	A	1prt	D	2bop	A	1apy	A	1hfh		1ped	A	1zym	A	1snc		1fua		8atc	A	1nhp	
1bnd	A	1ice	B	1prt	B	2bpa	2	1asz	A	1ihf	B	1pmi		2abh		1gtq	A	2abk		2cmd		1gcb	
1bpl	A	1ice	A	1ptd		2bpa	1	1axn		1iml		1pms		2arc	A	1pbx	A	9pap		1hvd		1pii	
1bpl	B	1ilk		1ptv	A	2cas		1beo		1iol		1pot		2bbi		3sdh	A	1thj	A	1gsa		2hpd	A
1bri	C	1irl		1ptx		2cpl		1bhm	A	1irs	A	1ppr	M	2fcr		1ash		1vid		1tag		3grs	
1bvp	1	1knb		1pvc	2	2end		1bmf	D	1iso		1psc		2fha		1vsd		2ayh		2acq		1gph	1
1bw4		1kny	A	1pvc	1	2gmf	A	1bmf	A	1ivd		1pud		2hpe	A	2fal		1gpc		1tca		1sat	
1cau	A	1kpb	A	1pyp		2gst	A	1bp1		1jac	A	1pue	E	2myr		8atc	B	2brd		1tah	A	1dnp	A
1cau	B	1kpt	A	1rbu		2kau	B	1bpy	A	1jer		1qap	A	2pfl		2hbg		2ak3	A	1quk		2pgd	
1cew	I	1l17		1rcb		2liv		1bro	A	1jpc		1qba		2pld	A	2mta	C	1nfp		2pia		6taa	
1chd		1lau	E	1reg	X	2mev	1	1btv		1jsw	A	1rai	D	2tbd		1sra		1pya	B	1poy	1	1dpg	A
1chk	A	1len	A	1rfb	A	2min	B	1bur	T	1jud		1rcy		2tys	B	1jcv		3pgm		1qor	A	1cow	A
1cks	B	1lfb		1rib	A	2nac	A	1cdq		1jvr		1rga		4rhn		2gdm		1din		1hmy		1byb	
1cmb	A	1lis		1rva	A	2ncm		1cem		1kap	P	1rgs		**COF**		1afb	1	1dhr		1nif		1smd	
1cns	A	1lki		1scm	C	2olb	A	1cex		1kaz		1rie				1mls		1gdo	A	1arv		8cat	A
1cnv		1lpe		1scu	B	2pii		1ckm	A	1kit		1rmd		5rxn		1phr		1cyd	A	1atp	E	1dpe	
1col	A	1lts	D	1ses	A	2ple	A	1cof		1klo		1rvv	1	1aaf		1esl		1bmt	A	2dld	A	1mmo	D
1com	B	1lts	A	1smn	A	2pol	A	1cpo		1kob	A	1ryc		1dtx		1hlb		1mrj		1pnr	A	1crl	
1cse	I	1lxa		1srs	A	2rsl	B	1cpq		1kuh		1ryt		1cdr		1jap	A	1ctm		1kif	A	1clc	
1ctn		1mal		1sva	1	2scp	A	1crk	A	1kve	B	1sei	A	1cea	A	1vhh		1nba	A	1mbb		1aoz	A
1cus		1mda	H	1svc	P	2sil		1csn		1kxu		1sft	A	1hcn	A	1bcf	A	1plq		2omf		3pmg	A
1cyu		1mhc	A	1svp	A	2tgi		1cyw		1l68		1shc	A	1pcn		1cyx		1tys		1rpa		2kau	C
1dlc		1mhl	C	1tam		2tmv	P	1def		1lbd		1sme	A	1fim		1obp	A	1dea	A	1fkx		4aah	A
1dpb		1mla		1tbr	R	2vil		1dek	A	1lbi	A	1stm	A	1umu		1std		1eny		2mnr		1pox	A
1dsb	A	1mml		1tfs		3cd4		1dhp	A	1lbu		1tcm	A	1mhl		3dfr		3fru	A	1ece	A	1aor	A
1dup	A	1mmo	G	1thv		3pga	1	1div		1lck	A	1tdt	A	9rnt		5p21		1ndh		1vsg	A	1sly	
1dyn	A	1mmo	B	1thx		3pte		1dkz	A	1lcl		1tf4	A	2psp		1rcf		2dri		1pea		1gof	
1ecp	A	1mol	A	1tii	D	3sic	I	1dor	A	1lid		1tfe		1put		1mka	A	1dih		1cdo	A	1trk	A
1ede		1msc		1tiv		3tgl		1dos	A	1lit		1tgx	A	2fd2		1rci		1qrd	A	2btf	A	1cyg	
1edg		1mse	C	1tlk		4gcr		1drw		1lrv		1uae		1fkj		1prr		2hhm	A	1hpm		1lcf	
1eri	A	1mut		1tnr	A	4rhv	3	1dxy		1lzr		1uby		2cdv		1fcd	C	1daa	A	1buc	A	1oac	A
1erw		1nal	1	1tpg		5tim	A	1eal		1mbd		1ucw	A	1msa		2prd		2prk		1ubs	B	2tmd	A
1esc		1nar		1trr	A	6fab	L	1ebp	A	1mhy	G	1ulp		1rtp		1cid		1nip	A	1nsc	A	8acn	
1etc		1nhk	L	1ttb	A	8ruc	I	1ecr	A	1msf	C	1uxy		1ccr		1dlh	A	2ebn		1pbe		1gpb	
1exg		1noy	A	1urn	A	8tln	E	1edh	A	1mty	B	1vcc		2hmz	A	1lfa	A	1tml		4xia	A	1bgl	A
1fbr		1omp		1vca	A	**TS97**	_	1emk		1nfa		1vhi	A	1sri	B	2stv		1han		1svb			
1fnf		1pba		1vhr	A	1aa6		1etp	A	1nfk	A	1vls		1bp2		1xnb		1scu	A	1oyc			
1ghr		1pbn		1vin		1aa8	A	1eur		1nkl		1vnc		2mad	L	2sas		1amp		1inp			
1gln		1pcr	H	1vmo	A	1ad2		1fbt	A	1nox		1vok	A	4fgf		1gky		2cyp		1chm	A		
1gpr		1pdn	C	1was		1ad3	A	1fib		1npo	A	1vsc	A	7rsa		1dlh	B	2ora		1fcd	A		
1hbq		1pgs		1xaa		1afr	A	1fro	A	1nsy	A	1wba		2phy		1isc	A	1sch	A	1oxa			
1hce		1pi2		1xyz	A	1agr	E	1gal		1nzy	A	1whi		2ccy	A	1tup	B	1ctt		1psd	A		

Table B. Three protein subsets used in the first part are shown with their PDB codes and chains.

Validation Proteins											
Val Set 1			Val Set 2			Val Set 3			Val Set 4		
Protein Name	Protein Length	Contact Number	Protein Name	Protein Length	Contact Number	Protein Name	Protein Length	Contact Number	Protein Name	Protein Length	Contact Number
1tgx	60	82	1xer	102	176	1aih	170	174	1ecr	305	372
1psc	69	103	1rga	104	150	1wba	171	318	1dor	311	521
2bbi	71	77	1msf	105	73	2fha	172	154	1nfk	312	513
1hcp	75	77	2pld	105	104	2fcr	173	275	1ppr	312	300
1iml	76	86	1jpc	108	174	1amm	174	323	1ucw	315	475
1cdq	77	111	1jer	109	159	1aoc	175	255	1ckm	317	504
1kve	77	85	1irs	112	157	1fro	176	194	2abh	321	551
1vcc	77	108	4rhn	115	144	1nfa	178	207	1pot	322	492
1nkl	78	72	1rmd	116	126	1pbw	184	184	1axn	323	367
1npo	81	146	1hfh	120	209	1fbt	186	265	1bpy	326	404
1pue	88	85	2pfl	121	183	1etp	190	249	1sme	329	619
1ihf	94	70	1whi	122	229	1ryt	190	199	1dxy	332	478
1who	94	135	1bur	123	131	1vok	192	284	1xel	338	564
1beo	98	107	1otg	125	116	1zxq	192	351	1got	339	697
2hpe	99	140	1oun	125	152	1shc	195	261	1aa8	340	526
			1eal	127	170	1vsc	196	314	1uxy	340	588
			1rie	127	208	1bhm	198	292	1xik	340	364
			1agr	128	117	1cex	200	333	1afr	345	391
			1cpq	129	113	1nox	200	220	1uby	348	339
			1lzr	130	181	1ebp	211	330	1pax	350	500
			1sei	130	190	1edh	211	371	1ped	351	698
			1lid	131	173	1lbu	214	301	1air	352	707
			1lit	131	207	1dkz	215	286	1kob	352	509
			1kuh	132	193	1hav	216	375	1eur	361	724
			1jac	133	252	1emk	220	400	1cem	363	571
			2tbd	134	185	1jud	220	298	1dos	369	596
			1cof	135	188	1akz	223	311	1pud	372	595
			1pms	135	163	1ad2	224	310	1kaz	378	608
			1jvr	137	50	1occ	227	266	1crk	380	563
			1anu	138	246	1lrv	233	275	1ofg	381	570
			1vhi	139	175	1lbd	238	227	1sft	382	640
			1lcl	141	228	1dek	240	286	1mty	384	387
			1stm	141	231	1zym	247	323	1ivd	388	858
			1tfe	142	160	1fib	249	427	2tys	396	721
			1occ	144	58	1tdt	256	432	1iso	414	660
			1rai	145	204	1yas	256	414	1gtm	417	691
			1vls	146	102	1occ	261	207	1uae	418	843
			1def	147	267	1rgs	264	385	1gnd	430	652
			1div	149	172	1ako	268	407	1gpl	432	793
			1gds	151	126	1nzy	269	354	1pmi	440	751
			1rcy	151	275	1nsy	271	339	1ad3	446	655
			1ulp	152	269	1drw	272	411	1alk	449	860
			1mbd	153	125	1kxu	276	280	1bp1	456	699
			1rvv	154	233	1bro	277	433	1jsw	459	612
			1btv	159	212	1iol	284	354	1bmf	467	753
			1cyw	159	253	1xsm	288	321	1kap	470	849
			1xjo	160	233	1qap	289	415	1bmf	487	770
			1apy	161	191	1ryc	291	382	1gow	489	790
			2arc	161	238	1dhp	292	483	1asz	490	689
			1klo	162	303	1xva	292	391	1occ	514	633
			1l68	162	167	1csn	293	389	2myr	519	169
			1lck	164	236	1gym	296	485	1vnc	576	928
			1mhy	167	131	1lbi	296	488	1gal	581	1108
						1cpo	299	407	1tf4	605	1041
									1tcm	686	1291
									1aa6	696	1276
									1kit	757	1503
									1qba	863	1575
									1alo	908	1790

Table C. Proteins in the validation set TS97. Contact numbers are obtained using the contact definition in section 2.3.3 in the second part of the thesis and contacting residues whose sequence separation is less than four residues are not included.