INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

Hyderabad

Graph spectral approach for domain identification in proteins

A thesis submitted in partial satisfaction

of the requirements for the degree Master of Science by Research

in Bioinformatics under the guidance of Dr. Nita Parekh

by

Hari Krishna Yalamanchili

(200761004)

Center for Computational Natural Sciences and Bioinformatics

harikrishna@research.iiit.ac.in

2009

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

Hyderabad

I certify that the work contained in this thesis, titled "**Graph spectral approach for domain identification in proteins"** by Hari Krishna Yalamanchili has been carried out under my supervision and in my opinion, is fully adequate in scope and quality as a dissertation for the degree of Master of Science by Research in Bioinformatics.

Date:

Dr. Nita Parekh (Principal Advisor)

(Assistant Professor, CCNSB, IIIT-Hyderabad)

International Institute of Information Technology, Hyderabad

2009

ACKNOWLEDGMENT

I would like to take this opportunity to thank everyone who played a part in this research effort. First and foremost, I would like to thank my advisor, Dr. Nita Parekh for her great guidance and timely support over the past two years. I am greatly thankful that she took me as a student, taught me the research methodology, guided me in choosing interesting and influential research topic and encouraged me when I was stuck. Her support and guidance was a constant source of inspiration for me towards the completion of the thesis work. She left no stone unturned to make this thesis a success.

I want to thank the Center for Computational Natural Sciences and Bioinformatics at the International Institute of Information Technology, Hyderabad for giving me support in various forms throughout my research and that helped me in finishing my work. I gratefully acknowledge all the faculty members of CCNSB Dr. Abhijit Mitra, Dr. Harjinder Singh, Dr. Savita, Dr. Rameshwar, Dr. Gopal, Dr. Deva and Dr. Prabhakar for the training and constant feedback, which gave me an assessment of my work.

I would like to thank all my friends, who have made my stay at IIIT Hyderabad a pleasure. Ashwath for giving me all the support I needed in tough times and a word of caution when it was most required. Sreekanth, Karunakar and Nagarjuna for their valuable suggestions and theories. Bharghava for the various discussions we had ranging from music to life. Padmini and Srilakshmi for their confidence that things will work out fine. Ruchi and Raina for all the support and help they have given me. This would not have been possible without you guys.

I would also like to thank my family members, whose words of care and concern never fail to have an effect on me. This thesis wouldn't have been possible without the constant love and support of my family.

PUBLICATIONS

- Hari Krishna Yalamanchili and Nita Parekh, (2009). "*Graph Spectral Approach for Identifying Protein Domains*". LNCS, Proceedings of International Conference on Bioinformatics and Computational Biology, BICoB 2009: 437-448.

- Hari Krishna Yalamanchili and Nita Parekh, (2009). "*Graph Spectral Approach for Identifying Protein Domains*". Proceedings of National Symposium on Cellular and Molecular Biophysics, NCMB-2009: pp108.

- Ruchi Jain, Hari Krishna Yalamanchili and Nita Parekh "*Identifying Structural Repeats in Protein using Graph Centrality Measures*". To appear in the Proceedings of World Congress on Nature and Biologically Inspired Computing (NaBIC'09).

TABLE OF CONTENTS

LIST OF FIGURES:

LIST OF TABLES

ABSTRACT

Proteins are the most complex organic compounds synthesized in nature and are an important class of biomolecules that serve as essential building blocks of the body. Proteins are organized into structural/functional units like motifs, domains and folds. So far, efforts to understand protein evolution and its functions have focused on domains that can fold into a stable three-dimensional structure independent of the rest of the protein chain and perform a unique function conserved over the evolution. Genetic recombinant techniques allow reorganization of domains at the amino acid sequence level (such as circular permutations). This along with swapping and insertion of domains are responsible for new protein functions and evolution. Protein domains are very useful in analyzing the mechanisms of protein folding and their stability and structural transformations in various conditions. This can be done reliably if a multi-domain target is considered only in terms of its constituent domains. The underlying goal is to reduce a complex protein structure to a set of simpler yet structurally meaningful units, each of which can be analyzed independently. Being the basic units of protein folding, function and evolution, the identification and analysis of domains forms the first step in understanding the protein functional and structural aspects.

Although the boundaries of a domain can be determined by visual inspection, there is an urgent need for the development of accurate methods for automatic domain identification as the number of solved protein structures is increasing rapidly. This problem of dividing a protein structure into domains is not yet solved efficiently due to the lack of an unambiguous definition of a domain. Existence of noncontiguous domains adds to the difficulty in developing an automated solution for domain partitioning. Though several methods have been proposed to predict domains they all have notable limitations. Even current reliable methods such as DomainParser, based on graph partitioning approach, do not distinguish between single and multi-domain proteins. Another recent graph-spectral approach by Sistla *et al* is unable to identify noncontiguous multi-domains and also does not distinguish between single and multi-domain proteins. With about 25% of multi-domain proteins being noncontiguous, there exists a need for reliable domain identification methods.

Here we present a simple method based on graph spectral properties for automatic domain prediction from its 3-dimentional structure. We show that graph properties such as density and interaction strength apart from spectral analysis helps in domain identification. Since most single domains are typically less than 300 amino acids long, our first screening for single domain proteins is done on the basis of length. Next we show that the density shows a gradual decrease with the increase in the length of the protein, thus providing a useful parameter for domain identification. It is a common observation that the interactions between the amino acids are higher within a domain (intra-domain interactions) than across the domains (inter-domain interactions). We exploit this feature for identifying the domain boundaries and also for distinguishing between single and multi-domain proteins. In general, domains are identified either by: (i) Top-down approach, i.e., starting with the whole structure partition it iteratively into smaller units. (ii) Bottom-up approach, i.e., defining very small structural units and assembling them into domains, or (iii) a combination of both. In the proposed approach for domain identification, both, top-down and bottom-up approaches are implemented. The algorithm proceeds by first decomposing the protein contact network into compact structural modules and then assembling them into domains. These interactions and the topological details of protein structures can be effectively captured by a protein contact network, constructed by considering each amino acid as a node, and with an edge drawn between two nodes if the C_α atoms of the amino acids are within 7Å. Newman's community detection approach proposed for social networks is employed here to identify domains in protein structures as we observe similarities between the two problems. It is particularly suitable for domain identification problem as one may not have *a priori* knowledge of the number of structural domains in a protein. The basic principle of this approach is to divide the vertices into two groups so as to minimize the number of edges running between the groups, which is in agreement with the basic feature of domains that inter-domain connections are more in number than connections across domains. We have implemented this approach on protein contact networks and analyzed the eigenvectors of the largest eigenvalue of the modularity matrix (which is a modified form of the Adjacency matrix).

First, a given protein structure is classified as a single domain protein based on length threshold (≤ 300 a.a,) and Interaction strength of the first split (≥ 15) in the Newman's modularity based algorithm. For multi-domain proteins, the protein contact network is decomposed into compact

structural modules by using a quality function called “modularity” to identify the optimal divisions of the network. The subdivision with the largest value of modularity is considered as the best natural subdivision of the network. Next we identify complete compact structural domains by implementing a hierarchical agglomerative algorithm by Clauset *et al* on these structural subgroups and then join them based on the interactions between them. This agglomeration is done till the best modularity value of the network is obtained. The analysis has been carried out on a set of 100 proteins (comprising of 50 single and 50 multi-domain proteins many of which are non-contiguous) and belonging to different structural classes, *viz*., α, β, α/β and α+β. The results on this dataset are compared with annotations in CATH database, wherein the domain boundaries are assigned manually based on the analysis of results derived from a range of algorithms and also with the output of DomainParser, a web-based domain prediction program based on graph partitioning approach. Prediction accuracies are also reported for the dataset of 55 proteins (21 multi domain and 34 single domain) compiled by Jones *et al* for comparison with other domain identification programs. A very good agreement is obtained not only in the number of domains predicted but also in the prediction of domain boundaries as compared to the annotation in CATH database. The accuracy of our approach on Jones *et* al dataset is 91% and the overall accuracy is 93%.

Chapter 1

Introduction

1.1 Proteins: Building Blocks of the Body

Proteins are complex organic compounds constructed from unbranched chains of amino acids and joined together by peptide bonds between the carboxyl and amino groups of adjacent amino acid residues, hence also known as polypeptides. These are an important class of biomolecules that serve as essential building blocks of the body. Proteins were first described and named by the Swedish chemist Jons Jakob Berzelius in 1838. They perform various functions such as catalysis, transport and storage molecules, immune protection, generate movement, transmit nerve impulses and control growth and differentiation. They function as enzymes that regulate the manner in which genes direct production of other proteins, catalyze biochemical reactions and act as receptors for hormones and other signaling molecules. Proteins make up much of the cellular structures, *viz.*, skin, fingernails, etc. Proteins also have mechanical functions, for e.g., actin and myosin in muscle and the proteins in the cytoskeleton, which form a system of scaffolding that maintains cell shape. They can interact and bind with one another and other biological macromolecules to form complex assemblies (Freeman and Stryer, 2002). The three-dimensional structures of proteins have evolved to carry out these functions efficiently and under precise control. Thus the spatial (three dimensional) organization of proteins is key for understanding their role and mode of action. Protein structure is organized in four hierarchical levels: Primary, Secondary, Tertiary and Quaternary structure as shown in the Figure 1.1.

Primary: The sequence of the different amino acids is called the primary structure of the peptide or protein. Counting of residues always starts at the N-terminal end (NH2-

group), which is the end where the amino group is not involved in a peptide bond. The primary structure of a protein is determined by the genes corresponding to the protein. The sequence of a protein is unique to that protein, and defines the structure and function of the protein. It is considered as a one dimensional structure. Primary structure is also called the "covalent structure" of proteins because, with the exception of disulfide bonds, all of the covalent bonding within proteins defines the primary structure as shown in Figure 1.1(a).

Secondary: It is the locally occurring structure in proteins and is mainly formed through hydrogen bonds between backbone atoms. There are three types of secondary structural elements:

Alpha helices: The backbone of an alpha helix is arranged in a spiral (similar to that seen on a cork screw) and is stabilized by hydrogen bonds between the carbonyl oxygen of one amino acid and the backbone nitrogen of a second amino acid located four positions away as shown in the Figure 1.1(b).

Beta-sheets: The backbone of a beta sheet is arranged in pleated manner. A minimum of two strands is required to define a beta sheet. The beta sheet is stabilized by hydrogen bonds between the carbonyl oxygen of an amino acid in one strand and the backbone nitrogen of a second amino acid in another strand as shown in Figure 1.1(b’). Beta sheets can be either parallel or anti-parallel. If the amino terminal residue of each strand "points" in the same direction the sheet is considered parallel. Anti-parallel sheets have the amino termini "pointing" in opposite directions.

Turns: Turns are ‘U’ shaped, typically four residues long segments with the distance between any two C_α atoms (< 7 Å). These residues are not involved in a regular secondary structure element such as an alpha helix or beta sheet.

Alpha-helices and beta-sheets are preferably located at the core of the protein, where as turns prefer to reside in outer regions.

Tertiary: The tertiary structure is the native three dimensional configuration of the protein under given environmental conditions. It describes the packing of alpha-helices and beta-sheets with respect to each other and whole polypeptide chain as shown in Figure 1.1(c). Disulfide bonds and hydrophobic interactions between amino acid side chains are responsible for the stabilization of the tertiary structure of the protein. It is important for the biochemical function of the protein.

Quaternary: The quaternary structure is the interaction between several chains of peptide bonds as shown in Figure 1.1(d). The individual chains are called subunits. The individual subunits are usually not covalently connected, but might be connected by a disulfide bond. It only exists if there is more than one polypeptide chain present in the protein complex. The quaternary structure is stabilized by the same range of interactions as the tertiary structure. Complexes of two or more polypeptides are called multimers. Specifically, it is called a dimer if it contains two subunits, a trimer if it contains three subunits and a tetramer if it contains four subunits and so on. It describes the inter chain interactions and spatial organization of the multiple chains.

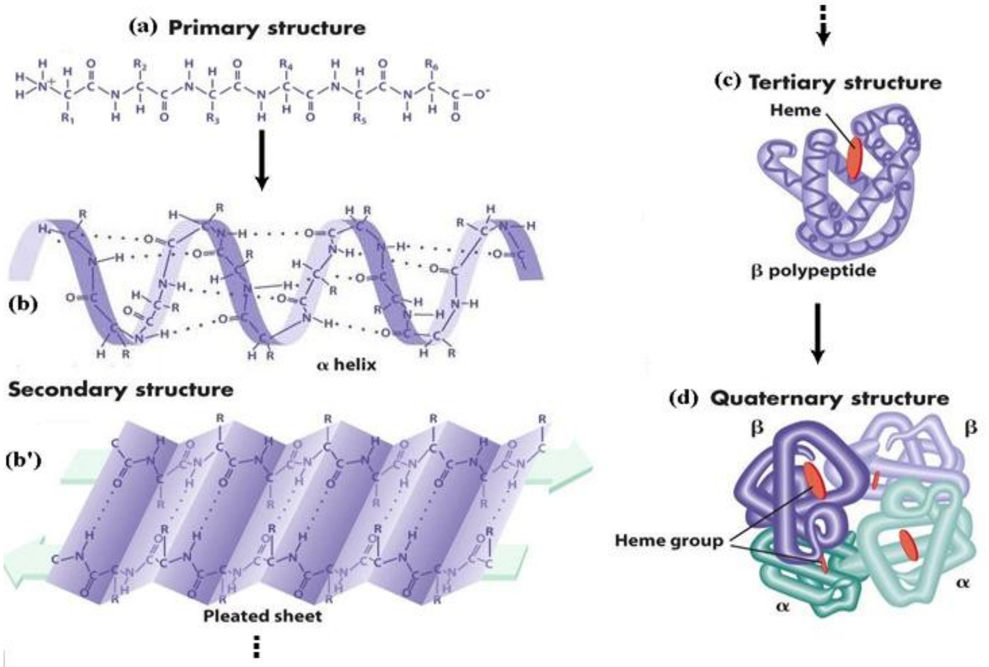


Figure 1.1: (a) Primary structure, (b) α helix, (b') β sheet, (c) Tertiary structure, (d) Quaternary structure

Motif refers to a small specific combination of secondary structural elements. These elements are often called super secondary structures. These structures can be as simples as, alpha-alpha (two alpha helixes linked by a loop), Beta-Beta (two beta-strands linked by a loop), Beta-alpha-Beta (Beta-strand linked to an alpha helix that is also linked to other beta strand, by loops) or more complexes structures, like the Greek key motif as shown in the Figure 1.2 or the beta-barrel. Even if the spatial sequence of elements is the same in all instances of a motif, they may be encoded in any order within the underlying gene. Protein structural motifs often include loops of variable length and unspecified structure, which in effect create the "slack" necessary to bring together (in space) two elements that are not encoded by the immediately adjacent DNA sequences in a gene.

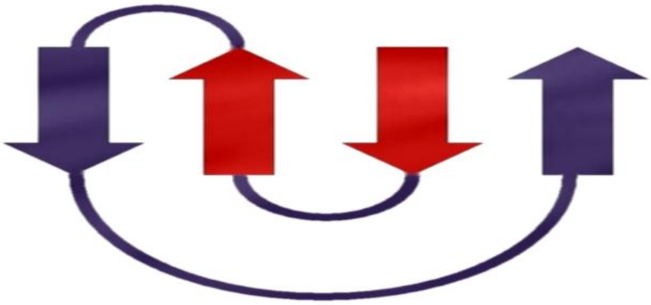


Figure 1.2: Greek key motif

Domains are stable, independently folding, compact structural units within a protein, formed by segments of the polypeptide chain, with relatively independent structures and functionally distinguishable from other regions. Thus protein domains may be considered as elementary units of protein structure and evolution, capable of folding and functioning autonomously. The tertiary structure of many proteins is built from several domains. Protein function is based on its constituent domains.

Fold refers to a global type of arrangement of the two well-defined secondary structural units of helices and sheets that are abundant in proteins. Protein structures are classified upon the alpha-helix and beta sheet and their general topological properties like bundle or barrel respectively. Each different topology is considered as a fold. Though many functions are connected to specific types of folds, always folds are not linked to any functional classification. In case of convergent evolution the actual fold of a protein is only a means to form a stable structural scaffold. On this scaffold, active sites and other functional properties are added. The existing folds are the result of their evolutionary history.

The shape into which a protein naturally folds is known as its native conformation (Ghelis and Yon, 1979). Although many proteins can fold unassisted, simply through the chemical properties of their amino acids, others require the aid of molecular chaperons to fold into their native states (Ostermeier and Benkovic, 2000). Proteins are not entirely rigid molecules they shift between several related conformations while they perform their functions. Thus protein structures are vital for their respective functions. Understanding proteins structural properties, their relation to function, folding kinetics, relevance of specific residues, and collection of residues such as active sites, domains, etc. are of considerable importance.

1.2 Importance of Protein Domain Identification

The concept of domains was first proposed by Wetlaufer in 1973. He observed regions with more atomic interactions within a protein structure and defined them as stable units of protein structure that could fold autonomously, and later termed them as domains. The definition of protein domains varies widely across the discipline of biology. Domains can be defined as the autonomous structural units that are 1) compact (Richardson, 1981), 2) stable, 3) contain a hydrophobic core, 4) can fold independently of the rest of the protein (Wetlaufer, 1973), 5) occur in combinations with different domains, and 6) perform a specific function (Bork, 1991). The domains are considered as the fundamental units of tertiary structures. Each domain contains an individual hydrophobic core built from secondary structural units connected by loop regions. The packing of the polypeptide is usually much tighter in the interior than the exterior of the domain producing a solid-like core and a fluid-like surface (Zhou *et al*, 1999). In fact, core residues are often conserved in a protein family, whereas the residues in loops are less conserved, unless they are involved in the protein's function. Domains have limits on sizes (Savageau, (1986) and vary in length from 36 residues in E-selectin to 692 residues in lipoxygenase-1(Jones *et al*, 1998). But the majority (~ 90%) have less than 200 residues (Siddiqui and Barton, 1995) with an average of approximately 100 residues (Islam *et al,* 1995). Very short domains (< 40 residues), are stabilized by metal ions or disulfide bonds. Larger domains (> 300 residues) are likely to have multiple hydrophobic cores (Garel, 1992). Figure 1.3 shows the domain decomposition of the protein papain. The segments of the protein structure encapsulated by the circles represent the two domains.

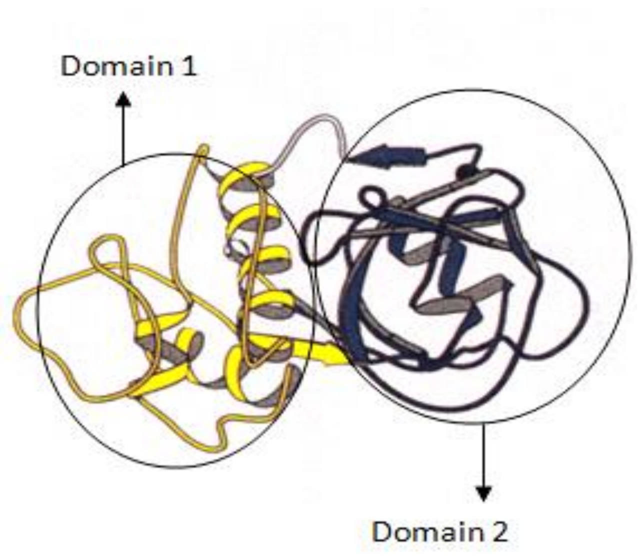


Figure 1.3: Domain assignment of the protein Papain.

Typically, in large protein molecules, several domains aggregate together to form multi-domain and multi-functional proteins with a vast number of possibilities (Sowdhamini and Blundell, 1995). In a multi-domain protein, each domain may function in association with its neighboring domains or each domain may have a separate function to perform for the protein. Different proteins containing the same domain content may have different functions because of the difference in the order of arrangement of the domains. This has been shown by the fact that the alignment of sequences containing similar domains, but in different orders, can result in poor and possibly misleading alignments. However alignment of the shared domains if extracted from the parent sequence may reveal a high level of sequence similarity, demonstrating an evolutionary link between the domain sequences.

Genetic recombinant techniques allow reorganization of domains at the amino acid sequence level (such as circular permutations). This along with swapping and insertion of domains are responsible for new protein functions.

- Domain swapping: It is a mechanism of forming oligomeric assemblies (Bennett, 1995). In domain swapping, a secondary or tertiary element of a monomeric protein is replaced by the same element of another protein. Thus resulting in structural and functional evolution of that protein.

- Domain Insertion: Not only do domains recombine, but there are many examples of a domain being inserted into another. This results in non-contiguous domains. In Figure 1.4 (a), we can clearly see that as we move from the N-terminus to the C-terminus the chain runs forming a part of the first domain (A1), moves to the second domain (B) and then back into the remaining part of the first domain (A2). Sequence or structural similarities to other domains demonstrate that homologues of inserted and parent domains can exist independently. An example is that of the 'fingers' inserted into the 'palm' domain within the polymerases of the Pol I family (Russell, 1994).

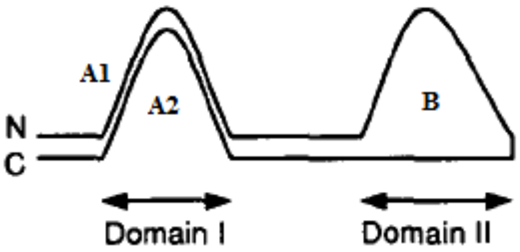


Figure 1.4: Non-contiguous domain (Siddiqui and Barton, 1995).

The concept of protein domains is very useful in analyzing the mechanisms of protein folding (Cunningham and Agard, 2003) and their stability and structural transformations in various conditions (Ahmad *et al*, 2004, Janin and Wodak, 1983). The functional (catalytic and ligand binding) sites of protein molecules are frequently located at the interfaces between their domains (Janin and Wodak, 1983). The identification and analysis of domains forms the first step in understanding the functional and structural aspects of proteins. As each domain has its unique function that contributes to the overall function of the protein, identification of domains and understanding their function is of primary importance. Furthermore domains can fold independently and form a distinct structural unit. Thus structural classification databases such as SCOP (Murzin *et al*. 1995) and CATH (Orengo *et al*. 1997) have their first level of classification only based on domains. The underlying goal is to reduce a complex protein structure to a set of simpler yet structurally meaningful units, each of which can be analyzed independently. Fold recognition methods performs more reliably if a multi-domain protein is considered in terms of its constituent domains rather than the whole chain (Jones & Hadley, 2000). Although the boundaries of a domain can be determined by visual inspection, development of an automated method is not straightforward, especially when domains are discontinuous or loosely associated (Sowdhamini and Blundell, 1995). Hence domain identification has been an important problem in protein structural analysis and various approaches have been proposed that vary with each research group using a unique set of criteria to define a domain (Swindells, 1995). Thus the problem of dividing a protein structure into domains is not yet solved efficiently and accurately due to the lack of an unambiguous definition of domain. Though several methods were proposed to predict domains they all have notable limitations. Even current reliable methods like DomainParser, based on graph partitioning approach, fails to identify single domain proteins and the graph-spectral approach by Sistla *et al* (2005) fails to identify not only single domain proteins but also non-contiguous domains, while about 25% of multi-domain proteins are non-contiguous. As a result, widely used databases of domains such as SCOP and CATH extensively rely on human expertise for domain assignments. However, with increasing number of protein structures in the PDB databank (Berman *et* *al*, 2003), it is no more possible to keep the domain databases up to date. There is apparently a need for computationally efficient, reliable and fully automated methods for domain identification.

1.3 Literature survey

The study of protein domain identification started nearly 40 years ago by Wetlaufer (1973) by visually inspecting the X-ray structures of lysozyme and papain. He pointed out the regions with extensive atomic interactions in a protein structure as domains. Later on many automated computational methods have been proposed for predicting domains both at the sequence and structural level. The approaches for domain identification based on sequence information are not very accurate, especially in the case of non-contiguous domains. However, besides these limitations, the sequence-based approaches are useful when no 3-dimentional structure of protein is available. A recent comprehensive review of computational approaches for domain identification is given by Stella and Ilya (2007). Below we briefly discuss some of the important methods for domain identification.

1.3.1 Sequence based methods

It has been observed that the sequences with substantial similarities in sequence (> 30% identity) share common domains that possess a common fold and thus usually share similarities in function (Doolittle, 1995; Ponting and Russell, 2002). The basic idea of all the sequence based methods is that the sequences containing similar domains have good alignment. These methods use an alignment approach where domains are identified by aligning the target sequence/secondary structure against sequences/secondary structures in the domain classification database with known domain boundaries (Marchler-Baueret *et al*, 2003). Some *ab initio* methods, such as tertiary structure folding approaches assign domain boundaries to a given sequence based on protein folding simulations (George and Heringa, 2002). One drawback of these approaches is that they are computationally intensive. Moreover these methods simply assign each conserved block (segment) in the multiple sequence alignment to a separate domain, ignoring the situation of non-contiguous domains. A few well-known sequence-based approaches are given below.

Pfam (1997)

Pfam is a database of protein domain families and their multiple sequence alignments (Sonnhammer *et al*, 1997). Here the domains are identified as parts/stretches of sequence, found in multiple protein sequences i.e. conserved across multiple sequences. This can be observed by constructing MSA (Multiple Sequence Alignment). Pfam maintains the quality of domain definitions on one hand and completeness on the other hand by having two parts A and B. Pfam-A is manually curated and contains well-characterized protein domain families with high quality alignments, which are maintained by using manually checked seed alignments and HMMs to find and align all members. Pfam-B contains sequence families that generated automatically by applying the automated HMMs to cluster and align the remaining protein sequences after removal of Pfam-A domains. One can submit a query sequence to Pfam and search against various protein families, and if a good similarity is found with members of any family, the domain organization in the query sequence can be identified. The database is available at: http://pfam.sanger.ac.uk/.

ProDom (2000)

ProDom is a domain database containing all protein domain families automatically constructed by clustering homologous segments generated from SwissProt and TrEMBL sequence databases (Corpet *et al*, 2000) (Bairoch and Apweiler, 1997). ProDom is built by automatically clustering the domain families using the MKDOM2 program (Gouzy *et al*, 1999) which is based on recursive PSI-BLAST searches. Firstly structural domain families from SCOP (Conte *et al,* 2000) are used to cluster homologous domains using PSI-BLAST (Altschul *et al*, 1997). The domain families are selected from SCOP on the basis of the following criteria: (i) length homogeneity (the shortest sequence can be at most 25% shorter than the longest); (ii) sequence homogeneity (below 450 PAM); (iii)

Minimum of two domains; (iv) No internal repeats and (v) length ≤ 500 amino acids. A position-specific scoring matrix (PSSM) is built for each of these families and used as a PSI-BLAST query in order to systematically cluster the homologous domains. ProDom is available at http://www. toulouse.inra.fr/prodom.html

DomPred (2002)

DomPred first predicts domains by searching the query sequence against the domain sequences from Pfam-A, and then reports if significant sequence matches are found (Marsden *et al*, 2002). PSI-BLAST (Altschul *et al*, 1997) is used for searching homologues. In cases where no clear homology exists to any known domain sequence, the query sequence is searched against a non-redundant sequence database, to identify significantly matching sequence homologues. These homologous sequences are then used to identify possible domain boundaries within the query sequence. When no sequence homologues are found, DomSSEA (Marsden *et al*, 2002) is used. It maps the predicted secondary structures of the query sequence with the observed secondary structure patterns of domains, whose 3-D structure is known. This is often the way in which a human sequence analyst will attempt to parse a protein into domains when homology-based approaches have been unsuccessful. DomPred is publicly available at http://bioinf.cs.ucl.ac.uk/ dompred/DomPredform.html.

Scooby-domain (2005)

Scooby-domain uses a multi-level smoothing window to predict the location of domains in a query sequence (George *et al*, 2005). A window size, representing the smallest domain size observed in the database to the largest domain size is used. Based on the window length and its average hydrophobicity, the probability that it can fold into a domain is found directly using the structure level domain representatives from the CATH domain database (Orengo *et al,* 1997). Probability matrix for a sequence is used to identify regions that are likely to fold into domains or are likely to be unstructured. The highest probability in the domain probability matrix represents the first predicted domain. The corresponding stretch of sequence for this domain is removed from the sequence and the resulting N and C termini fragments are rejoined and the probability matrix is recalculated as before. The second highest probability is then found and the corresponding sub-sequence removed. This continues till the entire query sequence is covered. It is publicly accessible at http://www.ibi.vu.nl/programs/scoobywww/.

CDD (2005)

The Conserved Domain Database (CDD) is a database of multiple sequence alignments representing protein domains conserved in molecular evolution (Marchler *et al*, 2005). It is a secondary database of publicly available domain alignment collections, such as Pfam (Sonnhammer *et al*, 1997), SMART (Jorg *et al*, 1998) and COG (Tatusov *et al*, 1997), and updated with domain hierarchies curated at NCBI. Domains are predicted using a service named CD-Search which runs reverse-position-specific BLAST (RPS-BLAST) [ref], a variant of the widely used PSI-BLAST algorithm by comparing the query protein sequences against databases of PSSM (position specific score matrices) derived from alignments in CDD. It is publicly available at http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml.

As all the sequence based domain prediction methods are based on the homologous domains already available and their alignment, these methods cannot predict new domains which are not reported in the reference database. Moreover it is difficult to predict domains correctly if the alignment is not significant or if the MSA is discontinuous. Proteins with same domain content can differ in the order of occurrence of domains resulting in poor and possibly misleading alignments. Thus use of structural information is necessary to improve the domain prediction reliability and accuracy.

1.3.2 Structure based methods

The most common principle in the structural partitioning of proteins into domains is based on interpretations of *contact density*, i.e., there are more residue-to-residue contacts within the same structural unit, rather than across different structural units. The first manual survey of structural domains was carried out nearly 40 years ago by visual inspection of the available X-ray structures (Wetlaufer, 1973). Rossman and Liljas developed the first algorithm to predict the domains from 3D-structures using C_α-C_α distance (Rossman and Liljas, 1974). This was followed by three other studies conducted by Crippen (1978), Rose (1979) and Wodak and Janin (1981). The implementation principle may be very different in different methods, but the underlying idea in almost all computational approaches is to find regions with high density of interactions and its extent of separation from the rest of the protein (Tsai and Nussinov, 1997). Methods for decomposition of structures into domains can be divided into two categories; those that are based on human expertise for evaluation of each structure i.e. by manual inspection and those that are completely automated.

1.3.2.1 Manual methods

SCOP

The Structural Classification of Proteins (SCOP) database is created by manual inspection and provides a detailed and comprehensive description of the structural and evolutionary relationships of all known protein structures (Conte *et al,* 2000). The first step in the classification of a protein is to divide it, where necessary, into domains because each domain is an autonomous functional unit. The basic idea that a domain is a region of a protein which has its own hydrophobic core and has relatively less interactions with the rest, making it structurally independent, is visually inspected by the expert groups for predicting domain boundaries. SCOP database is available publicly at: http://scop.mrc-lmb.cam.ac.uk/scop/

CATH

The CATH database employs a hierarchical domain classification of protein structures in the Protein Data Bank (Orengo *et al,* 1997). Only crystal structures with good resolution (< 4.0Å) are considered. NMR structures are also considered. To divide multi-domain protein structures into their constituent domains, a combination of automatic and manual techniques are used. After extracting highly resolved structures from PDB, similar proteins are grouped on the basis of sequence similarity. A representative structure is taken from each sequence family and is divided into domains using a consensus approach (Jones *et al*. 1998), i.e., if a given protein chain has sufficiently high sequence identity and structural similarity (i.e., 80% sequence identity, SSAP score ≥ 80) with a chain that has previously been chopped, the domain boundary assignment is performed automatically by inheriting the boundaries from the other chain. Otherwise CATH employs four automatic domain assignment methods - DETECTIVE (Swindells, 1995), PUU (Holm & Sander, 1994), DOMAK (Siddiqui and Barton, 1995) and CATHEDRAL (Redfern *et al*, 2007). If all four methods provide unambiguous domain boundaries for a particular protein, then the domains are automatically detected else domain boundaries are assigned manually. CATH database is available publicly at: www.cathdb.info/

1.3.2.2 Automated methods

Automated domain partitioning can be divided into two fundamental approaches (i) Top-down: starting from the entire structure and proceeding to partition it iteratively into domains; Rossman and Liljas (1974), PUU (Liisa Holm and Chris Sander, 1994), DOMAK (Siddiqui and Barton, 1995) etc. are few examples that use this approach. (ii) Bottom-up: defining very small structural units (typically each amino acid) and assembling them into domains; Crippen's method (1978), DETECTIVE (Swindells, 1995), etc. are few examples of this approach. Some methods use both approaches within their algorithm: first decomposition and then assembly or vice versa. Below a brief description some of the structure-based domain identification methods is given.

Method by Rossman and Liljas (1974)

The first systematic algorithmic survey of domain identification was performed by Rossman and Liljas on a set of 3D-structures using C_α-C_α distance maps to identify structural domains (Rossman and Liljas, 1974). The distance maps are generated by calculating the distance between all (i, j) pairs of C_α atoms in the proteins. Domains were identified as a series of short-range interactions along the diagonal. However this simple approach fails to identify non-contiguous and loosely packed domains.

Crippen's method (1978)

This is a hierarchical clustering algorithm using C_α-C_α distance maps (< 9Å) based on the assumption that stable conformation of the protein is largely due to long-range interactions of the residues which are distant in sequence (≥ 7 residues) (Crippen, 1978). In the first step, the protein chain is divided into segments - short stretches of polypeptide segments such that none of the residues have long-range interactions with any other residue within the same segment. These segments are then hierarchically clustered using contact density criteria – number of long-range interactions between the two segments normalized by the size of individual segments. The process is repeated iteratively by first joining segments in the most intimate contact, while last clusters have few contacts relative to the number of residues involved.

Wodak and Janin Algorithm (1981)

The method of Wodak and Janin (1981) uses surface area measurements based on atomic positions to give structural domains in proteins. Segments of the polypeptide chain making a minimum of interactions with the rest of the protein structure are identified on interface area scans. It detects only continuous domains made of a single stretch of polypeptide chain. The surface area is calculated from the atomic coordinates using geometrical algorithm of Lee and Richards (1971). Then the protein is scanned for the smallest interface area B, which is calculated as

$$B = A_1 + A_2 - A_{12} \tag{1.1}$$

where A_1 and A_2 are the accessible surface area of each of the two groups and A_{12} is the accessible surface area of the two together. The groups with least interface area B are reported as domains. For multi-domain proteins the same procedure is iterated. The limitation of this method is that it cannot predict single domain proteins and non-contiguous domains correctly.

PUU (1994)

The algorithm, Parser for Protein Folding Units (PUU), identifies domains by creating a C_α-C_α contact matrix using a cutoff distance of ≤ 4.0Å and then by splitting this matrix such that the strongly interacting residues are grouped together by using reciprocal averaging (Lisa Holm and Chris Sander, 1994). Bisection of this contact matrix is continued recursively for each of the resulting folding units until some limit on unit size is reached. The following filters are employed for accurate prediction of domains: (1) domains size ≥ 40 so that units < 80 residues are never cut, (2) highly flexible units are always cut, (3) β-sheets are never cut, (4) a cut is acceptable if both resulting units have a high globularity/compactness value, (5) a cut that produces a non-globular domain with less than 40 residues is accepted on condition that the larger domain in the cut will be split into two domains upon recursive application of the filters.

DETECTIVE (1995)

Swindells (1995) developed the method, DETECTIVE, for identification of domains in protein structures based on the idea that domains have a hydrophobic interior. Hydrophobic cores are solid-like rigid structures which are well-fitted in protein interiors with the side chains neatly interlocked and least exposed to solvents. These are identified using a set of rules based on solvent exposure, minimal size of a core, fraction of the spatially adjacent residues, etc. These hydrophobic cores are considered as the centers of the domains and grow by iteratively including residues within the spatial proximity of the core, until most of the residues are assigned. Isolated residues that are generated are removed from the assignment. In the final step the unassigned residues are assigned by extending domains to both ends of the structure and/or to the ends of the appropriate secondary structure. The drawback of this method is its dependence on proteins having hydrophobic cores thus limiting it mostly to globular proteins.

DOMAK (1995)

This method was proposed by Siddiqui and Barton (1995). It is based on the concept that the residues comprising a domain make more contacts between themselves (internal contacts) than they do to the rest of the protein (external contacts). Firstly, a protein contact network is constructed by considering the residues whose heavy atoms are within 5Å of interacting residues. The algorithm progresses by chopping the protein chain into two parts. A segment can consist of any number of residues, but the residues must form a continuous sequence along the chain: segment A that consists of residues 1 to i and segment B of residues (i + 1) to N, where N is the number of residues in the chain. The split value can then be calculated for $1 < i < N$ as $(\mathrm{int}_A \times \mathrm{int}_B)/\boldsymbol{ext}_{AB}$ where int_A is the number of internal contacts in A, int_B is the number of internal contacts in B, and ext_{AB} is the number of contacts between A and B. For position i with largest split value, the protein is separated into two domains A and B. For handling discontinuous domains the protein is scanned for two split positions, for instance, say x and y as shown in Figure 1.5. For a domain with N segments, $2N$ maximum split points need to be scanned, making it computationally very expensive.

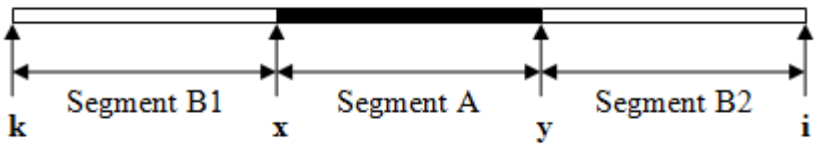


Figure 1.5: For discontinuous domains in which domain B is made of two segments B1 and B2.

Once the segments are obtained each segment is validated based on the following criterion (i) minimum domain size (MDS), 40, (ii) minimum segment size (MSS), 25, (iii) minimum split value (MSV), 9.5, and (iv) maximum allowed compactness (MAC), 2.5Å. The segments satisfying these criteria are reported as domains. Since MAC (maximum allowed compactness) is used as a validation filter, loosely packed domains (especially in fibrous proteins) are not identified accurately. It is freely available for download at http://www.compbio.dundee.ac.uk/ Software/Domak/domak.html.

Method by Sowdhamini and Blundell (1995)

The method proposed by Sowdhamini and Blundell (1995) clusters secondary structures in a protein, based on their C_α-C_α distances. Secondary structures (α-helix and β-strand) are identified based on the main chain hydrogen bonding patterns using the algorithm of Kabsch and Sander (1983). A proximity index, p_{ij}**,** is calculated for every (*i, j*) pair of secondary structure, which is the average distance of all C_α atom pairs between i^{th} and j^{th} secondary structural elements. Proximity index measures the strength of interactions between a pair of secondary structures. The calculated proximity indices are used to cluster secondary structural elements using the program KITSCH, which is a part of the phylogeny inference package PHYLIP (Felsenstein, 1985). Domains are then defined by grouping the clusters in the dendrogram such that the disjoint factor is greater than one (*df* >1). Disjoint factor measures the density of interactions between secondary structures within the domain relative to all the interactions of secondary structures in a protein.

Domain Parser (2003)

This algorithm, initially proposed by Xu *et al* (2000), uses a graph-theoretical approach to find the best partitioning of a given structure into two parts. Here the problem of domain decomposition is formulated as network flow problem, in which each residue of a protein is represented as a node in the network and each residue–residue contact as edge with a particular capacity depending upon the type of the contact. Two residues are defined to be in contact if the distance between their closest atoms is ≤ 4Å. The best partition of a given structure into two is obtained by finding the minimum cut of the network using the Ford-Fulkerson algorithm (Ford and Fulkerson, 1956). Minimum cut is the smallest set of edges which separate the vertices of the graph into two distinct sub-graphs. In Figure 1.6, the minimum cut (the dotted line in green) splitting the network into two is shown. The process is repeated iteratively for each of the resulting domains until the stopping criteria is met. In the second version of DomainPraser all the possible minimum cuts at each step are evaluated and then the best partition during a post-processing step is determined by checking several basic properties of the resulting domains (Guo *et al,* 2003). These properties include hydrophobic moment, the number of non-contiguous fragments, domain size, compactness, and relative motion of domains. The 'acceptable' range for each property as well as a set of stopping criteria is determined in advance using a neural network trained on the set of known structures. The limitation with this method is that it does not identify single domain proteins as in the

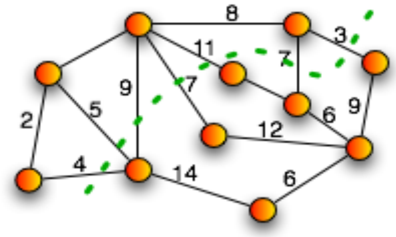


Figure 1.6 Minimum cut separating the nodes by minimizing the cross edge capacities.

very first step the method proceeds by splitting the network along the minimum cut. It is publicly accessible at http://compbio.ornl.gov/structure/domainparser/.

Method by Sistla *et al* (2005)

The graph-spectral approach given by Sistla *et al,* (2005) is based on the fact that the interactions among amino acids are higher within a domain than across domains. The central point of the algorithm is that the nodes (residues) of a domain have similar spectral parameters. Firstly protein graph is constructed by considering C_α atom of each residue as a node and interactions between them as edges. Two residues are said to be interacting if any their atoms fall within a distance of 6.5Å. Since eigenvector components corresponding to second smallest eigenvalue of the Laplacian matrix capture the clustering information, these eigenvector components have been used for domain prediction. The regions having similar eigenvector values are reported as domains, with domain boundaries identified as intersection points with its slope.

CATHEDRAL (2007)

CATHEDRAL employs a fast secondary-structure-based comparative method (using graph theory) to locate known folds and domains within a multi-domain context by aligning members of the target fold groups against the query protein structure to identify the closest relative and assign domain boundaries (Redfern *et al*, 2007). To increase the fidelity of the assignments, a support vector machine is used to provide an optimal scoring scheme. The search protocol is repeated in an iterative fashion until all recognizable domains have been identified. Since the predictions are based on structural similarity between the query and the database of target fold groups, domains with low structural similarity are not predicted accurately along with those that are not yet reported in the target fold groups.

Although the structure based domain prediction methods are superior to sequence based prediction methods, many of these methods suffer from the overcutting and undercutting of domains. Most of the recent structure-based domain identification approaches utilize the graph properties but have their limitations. For example, the spectral approach by Sistla *et al* is unable to identify single domains and non-contiguous domains. For a domain with *N* discontinuous segments, the DOMAK approach needs to scan *2N* maximum split points, making it computationally expensive. DomainParser constructs a network model with each atom defined as a node and edges drawn if two atoms are within 4Å distance. Compared to this, in the present approach proposed by us, the protein contact network is constructed using C_α atom as node thereby reducing the complexity of the network. Moreover, DomainParser is unable to distinguish between single and multi-domain proteins and the input to the program should be a multi-domain protein. These limitations suggest a need for efficient and reliable algorithm for domain identification in protein structures.

It is observed that the various approaches discussed above exploit different characteristics features of domains. To summarize, domains are structural units that are compact and stable entities that contain a hydrophobic core, and can fold independently of the rest of the protein. The domains may occur in different combinations in different proteins and perform unique functions. Another most extensively exploited feature of multi-domain proteins is that there exist more intra-domain interactions than inter-domain interactions. Minimum size of these structural units is ~ 40 with single domain proteins being typically < 300 amino acids long. Here we exploit some of the above mentioned properties in our approach for domain identification.

1.4 Organization of the Thesis

The thesis is divided into three chapters. In the first half of Chapter 2, a brief introduction of graph theory and various topological networks, *viz*., regular, random, small world and scale free networks and their properties is discussed. In the second half of Chapter 2, the graph properties and the spectral approach used for identifying protein domains is discussed in detail. In Chapter 3, we summarize the results of our analysis carried out on a dataset of 100 proteins and comparison of our results with the annotations in CATH database, a manually curated database, and with a web-based domain prediction program, DomainParser, based on graph partitioning approach. Finally the conclusions of our analysis are presented in Chapter 4.

Chapter 2

Materials and Methods

2.1 Introduction

Complex systems that are characterized by discrete constituents and their inter-relationships have been traditionally studied by modeling them as networks. Large complex networks arise in a vast number of natural and artificial systems. Ecosystems consist of species whose interdependency can be mapped into intricate food webs. Social systems may be represented by graphs describing various interactions among individuals. The Internet and the World-Wide-Web (WWW) are prototypical examples of self-organized networks emerging in the technological world. Large infrastructures such as power grids and the air transportation network are critical networked systems of the modern society. The living cell, with its organization and function, is the outcome of a complex web of interactions among genes, proteins and other molecules. Thus, a network could be anything that can be defined by a set of discrete elements (the vertices), and a set of connections (the edges) that link the elements, typically in a pair-wise fashion. By abstracting away the details of a problem, graph theory is capable of describing the important topological features with a clarity that would be impossible were all the details retained. As a consequence, graph theory has spread well beyond its original domain of pure mathematics, especially in the past few decades, to applications in engineering, operations research, computer science, sociology and biology.

Graph comparisons, quantitative characterizations, computation of topological indices, clustering and partitioning are some of the major computations which have yielded valuable results in various disciplines. With large amounts of high-throughput biological data now becoming available, graph theory provides a wide range of applications in the field of biology. Various problems have been addressed using graph theory approach to protein structures which includes pattern recognition, identification of folds, active sites, domains and critical residues. Graph theory has also been used in the analysis of various biological networks, e.g., gene regulatory networks, metabolic networks, protein-protein interaction networks, etc. to gain insight into molecular processes occurring in the cell.

Graph theory is a field of mathematics which studies objects as combinatorial structures called *graphs*. *Graph* is an abstract representation of a set of objects where some pairs of the objects are connected by links. The interconnected objects are represented by mathematical abstractions called *vertices or nodes*, and the links that connect pairs of vertices are called *edges*. For example, for the graph shown in the Figure 2.1, the vertex set is $V_A = \{1, 2, 3, 4, 5, 6\}$ and the edge set is $E_A = \{1\text{-}2, 1\text{-}5, 2\text{-}5, 2\text{-}3, 3\text{-}4, 4\text{-}6, 4\text{-}5\}$.

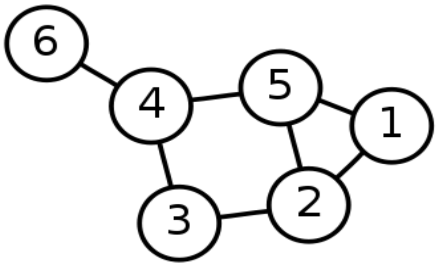


Figure 2.1: An example of a Graph showing numerically labeled nodes and edges between them.

A protein molecule is a chain of amino acids connected by peptide bonds and folds into a 3D structure which is responsible for its function. The 3-dimentional fold of a protein molecule is governed by number of non-covalent interactions such as hydrogen bonds, ionic interactions, Vander Waals forces and hydrophobic packing. For computational processing of such molecules an efficient data structure which can efficiently capture the structural properties of the molecule is desired and one such data structure is a Graph. In this thesis we use graph theoretic approach for the identification of domains in protein structures. Various graph properties used in the analysis are given below along with a brief introduction of various types of topological networks.

2.2 Network properties

Graphs are quantified by various local and global properties. Some of the important graph measures used in this study are briefly discussed below.

2.2.1 Degree

The degree (k_i) of a node i is the number of nodes to which it is directly connected (Costa *et al*, 2006). The degree signifies the connectivity of a node; the larger the degree the more important the node is and better is its connectivity in the network. Average degree K of a network with N nodes is defined as

$$K = \frac{1}{N}\sum_{i=1}^{N} k_i \qquad (2.1)$$

2.2.2 Characteristic Path Length

The characteristic path length of a network is defined as the average of the shortest path lengths, L_{ij}, between all i, j pairs (Costa *et al*, 2006). For example in Figure 2.2 there are three possible paths from START to END: Path 1 {A, B}, Path 2 {C, D, E} and Path 3 {F, G, E}. The shortest path in this case is Path 1 through edges A and B with $L_{start,\ end}$ = 2.

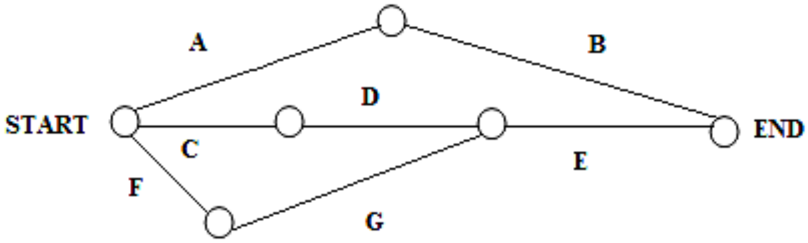


Figure 2.2: All possible paths between two points, START and END shown.

The characteristic path length (L) of a graph with N nodes is computed as the average of all such shortest paths, L_{ij}, between all possible (i, j) pairs:

$$L = \frac{1}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} L_{ij} \tag{2.2}$$

It is a measure of the global property of the network and indicates how well-connected a graph is, thereby reflecting the overall efficiency of the network, i.e., the ease with which information can be transferred to other entities in the network.

2.2.3 Graph Diameter

The maximum of the shortest path lengths between all possible vertices in a graph is defined as the diameter of a graph.

$$\textbf{Diameter } \boldsymbol{D = max\ L_{ij}, i, j} = \textbf{1… } \boldsymbol{N} \tag{2.3}$$

where L_{ij} is the shortest path length between residues i and j (Ganesh and Somdatta, 2007). This quantity gives the information regarding the compactness of the protein structure.

2.2.4 Clustering coefficient

The clustering coefficient C_i for a vertex v_i is given by the proportion of links between the vertices within its neighbourhood divided by the number of links that could possibly exist between them (Costa *et al*, 2006). In Figure 2.3(a), (b) and (c), the clustering coefficient calculation for the shaded node i is shown. Black lines represent actual edges connecting neighbors of i, while red edges are all possible edges between neighbors of i. Thus, the shaded node in the Figure 2.3(a) has 0 edges (no black edges) out of three possible edges (in red) between its three neighbors and so its clustering coefficient value is 0. In Figure 2.3(b) out of three possible edges there is one edge between the neighbors (one black edge) of node *i*, hence the clustering coefficient of the shaded node in this case is 1/3. Since all the three neighbours of the node *i* are connected with each other in Figure 2.3(c), its clustering coefficient is 1.

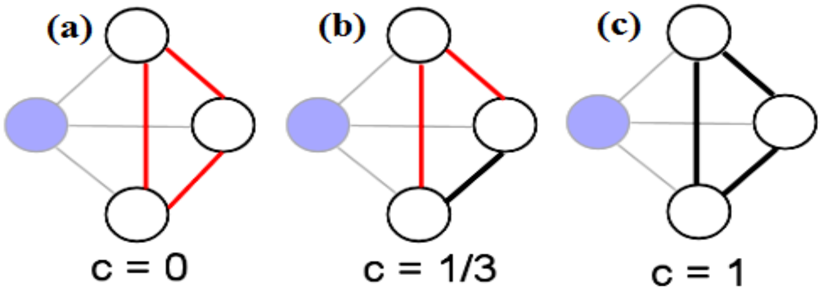


Figure 2.3: Illustration of the clustering coefficient calculation of the shaded node.

Numerically the clustering coefficient of the node *i* is computed as follows using the adjacency matrix, *A*:

$$C_i = \frac{1/2 \sum_{j=1}^{N} \sum_{k=1}^{N} A_{ij} A_{ik} A_{kj}}{{}^{k_i}C_2} \tag{2.4}$$

where k_i is the degree of the i^{th} node, *N* is the total number of nodes in the network and A_{ij} is 1 if nodes *i* and *j* are connected, else 0. The clustering coefficient for the whole netwrok is given by Watts and Strogatz as the average of the clustering coefficient for each vertex:

$$C = \frac{1}{N} \sum_{i=1}^{N} C_i \tag{2.5}$$

N is the total number of nodes.

2.2.5 Graph Density

The density of a graph is defined as the ratio of the total number of edges in the graph to the number of possible edges in a random graph with *N* vertices (Coleman *et al,* 1983). Graph density is a measure that indicates the compactness of a network.

$$\frac{\frac{1}{2}\sum_{ij} \boldsymbol{k_i k_j}}{\sum_{ij} \frac{\boldsymbol{k_i k_j}}{2\boldsymbol{m}}} \tag{2.6}$$

2.3 Mathematical Networks

Traditionally, networks have been modeled as either completely regular or completely random. However, the real world networks exhibit properties somewhere in between and are now classified as third type of networks, namely, small world networks. Below we briefly discuss the properties of various network topologies and show that protein structures exhibit small-world properties.

2.3.1 Regular Networks

In a regular (ordered) network, like a crystal lattice, each node has the same number of edges that join a small number of neighboring nodes in a tightly clustered pattern. Regular networks are simple, that is, locally the network "looks" the same everywhere, and this simplifies their analysis. The characteristic features of the regular networks are high values of both the characteristic path length (L) and the clustering coefficient (C). In Figure 2.4 (a) is shown a circular network with *N* (= 20) nodes and each node having degree k (= 4). Here each node is connected to its immediate and next immediate neighbours, i.e., to two neighbours on either side. In Figure 2.4 (b) is shown an example of a regular network on a 2-dimensional lattice where each node has degree $k = 4$. However, most real world networks exhibit a more complex connectivity pattern.

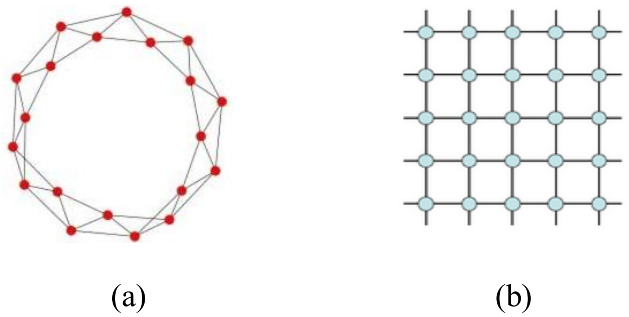


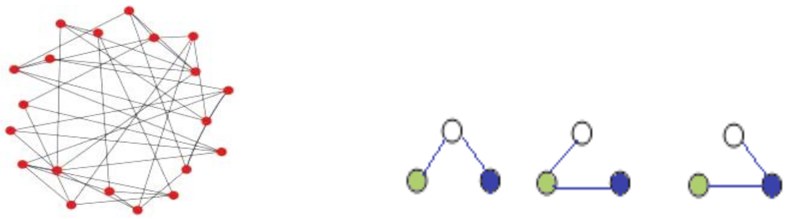
Figure 2.4: (a) regular lattice on a circle with *k* = 4, *N* =20. (b) 2-d regular lattice with *k* =4, N=25.

2.3.2 Random networks

It was earlier believed that real-world networks can be represented by random networks. Hence a lot of attention has been focused on analyzing the properties of random networks. Most commonly studied random graph model is the Erdos–Renyi model (Erdos and Renyi, 1959), There are two closely related variants of the Erdos–Renyi random graph model: (i) $G(n, p)$ and (ii) G(n, M). In $G(n, p)$ model, a graph with n nodes is constructed by randomly placing an edge between every pair of nodes with probability p, i.e., for all possible (i, j) pairs, a random number r is generated and the edge is placed if $r \geq p$ (Figure 2.5(a)). The probability of forming an edge between (i, j) pair is independent of the individual degrees of the i^{th} and j^{th} nodes. The process is continued till all the nodes are connected. In the $G(n, M)$ model, a graph is chosen uniformly at random from the collection of all graphs which have n nodes and M edges. For example, in the G(3, 2) model shown in Figure 2.5(b), each of the three possible graphs on three vertices and two edges are included with probability 1/3. The characteristic features of the random networks are low values of both the characteristic path length (L) and the clustering coefficient (C).

2.3.3 Small world networks

Most real-world networks appear to fall somewhere in between the two extremes discussed above: regular and random networks. Friendship networks are a good example of this in-between state. Since people meet most new friends through existing friends, the networks are locally ordered, i.e., if A knows B and B knows C, then A is more likely to know C than some other unknown person. The outcome of local ordering in such a network is that one individual's friends are more likely to know one another, a characteristic that is called "clustering."



(a) (b)

Figure 2.5: (a) Random Network with N=20 by *G*(*n, p*) model (b) *G*(*n, M*) model with *n*=3 and *M*=2.

Small world network is defined as a network in which most nodes are not neighbors of one another, but most nodes can be reached from every other node by a small number of hops or steps. The typical characteristic features of a small world network are that there are fewer nodes with large connections and many nodes with fewer connections as shown in the Figure 2.6. It exhibits lower characteristic path length and high clustering coefficient when compared to a random graph (with same number of nodes and edges). These networks are better resistant to random deletions of the nodes because vast majority are small degree nodes and the likelihood that a hub (a high degree node) would be affected is almost negligible. However, such networks can be quite sensitive to targeted attacks. Road maps, food chains, electric power grids, metabolite processing networks, protein contact networks, neural networks, voter networks, telephone call graphs, and social influence networks are few examples of small world networks.

Watts & Strogatz Small world model

Watts and Strogatz (1998) proposed the approach, called "random rewiring” to simulate small world networks by starting with an ordered network and introducing increasing amounts of randomness into it. Consider a regular ring lattice with N nodes, with each node connected to K nearest neighbours ($N >> K >> \ln(N) >> 1$), $K/2$ on each side.

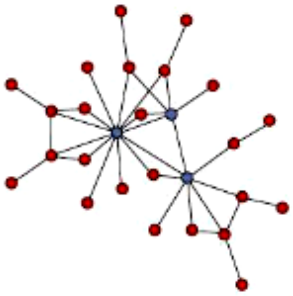


Figure 2.6: A representative example of a Small World Network.

In Figure 2.7 (reproduced from Watts and Strogatz, 1998), a regular network with $N = 20$ and $K = 4$ for every node on a ring is shown, i.e., every node is connected to its nearest and its next nearest neighbours on either side. Now all the edges are rewired with a probability p, one edge at a time. That is, for every edge between the nodes i and j with $i < j$, a random number r is generated; if $r < p$ the edge is detached from j and attached to another randomly chosen node k, where k is chosen with uniform probability from all possible values that avoid loops ($k \neq i$) and link duplication, i.e., there is no edge (i, j) with $j = k$. For small intermediate values of p, the network exhibits the small-world behavior, i.e., highly clustered like a regular graph, but exhibiting small characteristic path length, similar to a random graph (See Fig. 2.8).

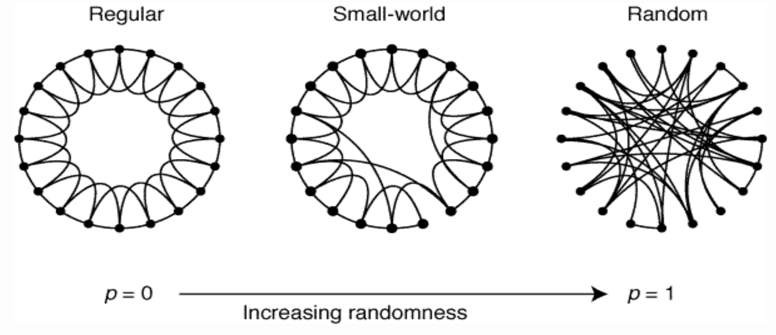


Figure 2.7: Watts & Strogatz Small world model (Watts and Strogatz, 1998)

The characteristic path length, $L(p)$, and the clustering coefficient, $C(p)$, for different values of the random rewiring probability p are plotted in Figure 2.8; $L(0)$ and $C(0)$ correspond to the path length and clustering coefficient values for p = 0. For $p = 0$, all the edges remain unchanged and the network remains regular. As is clear from the figure, for low p values (corresponding to almost regular network), the clustering coefficient is high and the path length is low. As p is increased, the graph becomes increasingly disordered and for $p = 1$, when all the edges have been rewired randomly, the resulting network is a random network. For p values close to 1, the clustering coefficient as well as the path length is low (similar to random network). For the intermediate range of p (0.01 - 0.1), shown by parallel lines in Figure 2.8, the clustering coefficient is high but the path length is low. This region is defined as exhibiting small-world behavior.

$$L_{regular} >> L_{small\text{-}world} \geq L_{random}$$

$$C_{regular} \geq C_{small\text{-}world} >> C_{random}.$$

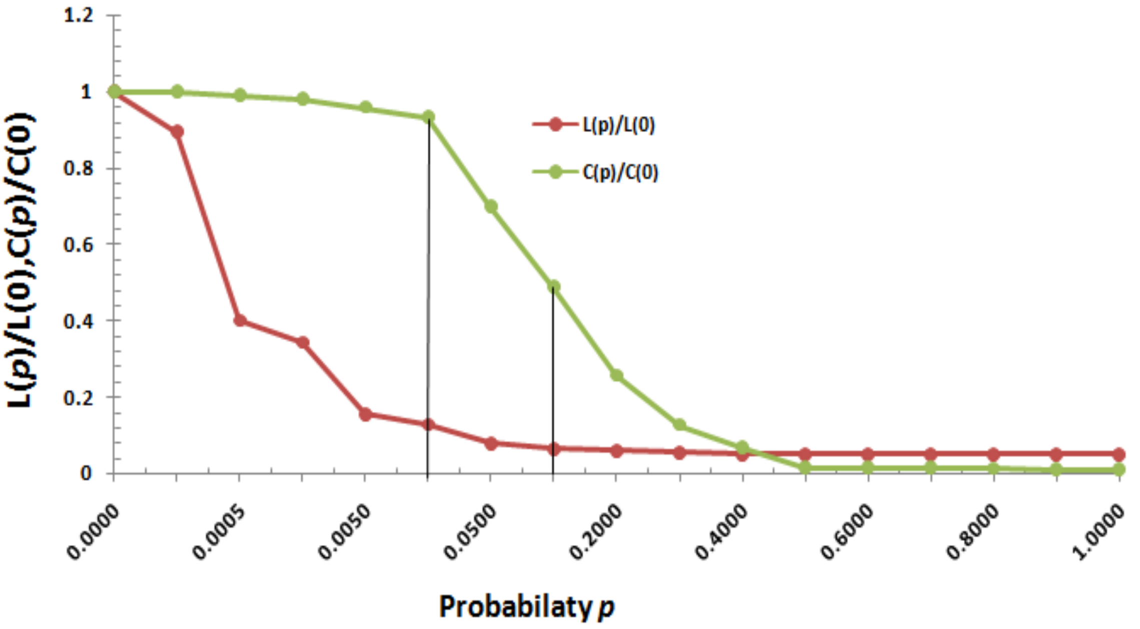


Figure 2.8 Characteristic path length $L(p)$ and Clustering coefficient $C(p)$ as a function of probability of rewiring p.

2.3.4 Scale-free networks

When Barabasi and his colleagues (Barabasi and Albert, 1999) mapped the connectedness of the World Wide Web, they found that its structure did not conform to the then accepted model of random connectivity. Instead, their experiment showed the existence of some nodes, which they called "hubs", with many connections as compared to others and that the network as a whole had a power-law distribution of the number of links connecting to a node which they called "scale-free." Thus a scale-free network can be defined as a connected graph or network with the property that the number of links '*k*' originating from a given node exhibits a power law distribution P$(k) \sim k^{-\gamma}$, where $P(k)$ is the fraction of nodes with the degree k and γ is a scaling constant whose values typically range between 2 and 3 ($2 < \gamma < 3$). Barabasi and Albert also proposed a method for the construction of scale-free networks, called "preferential attachment model" (Barabasi and Albert, 1999). According to this method, a scale-free network can be constructed by progressively adding nodes to an existing network and introducing links to existing nodes with preferential attachment so that the probability of linking a given node i is proportional to the number of existing links k_i that node has, i.e.,

$$P(Linked\ to\ a\ node\ i) \sim \frac{k_i}{\sum_j k_j} \tag{2.7}$$

The scale-free networks generated by the preferential attachment model also satisfy the small-world properties (Barabasi and Albert, 1999) like low characteristic path length and high clustering coefficient when compared to the corresponding random graphs. Like small-world networks, scale-free networks are also resistant to random removal of any node in the network. However in recent studies Michael *et al* (2008) have shown that it is possible to construct scale-free networks that are highly assortative but not small world. A few examples of scale free networks are social networks, protein-protein interaction networks, semantic networks, sexual partners in humans which affect the dispersal of sexually transmitted diseases, collaboration networks of movie actors in films and many kinds of computer networks including the World Wide Web. The small world model proposed by Watts and Strogatz also exhibits scale-free behavior. In Figure 2.9 a log-log plot of P(k) *vs* k is plotted for the Watts and Strogatz model with 1000 nodes and probability of rewiring $p = 0.06$. A straight line fit (with exponent $\gamma = 2.04$) confirms the power law distribution.

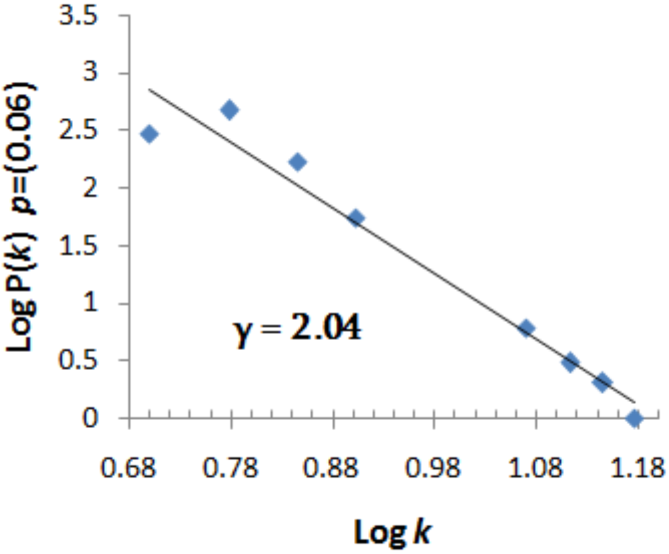


Figure 2.9: log-log plot of P(*k*) *vs. k* demonstrating the power law distribution

Apart from the scaling law for the degree distribution, there are two more scaling laws defined for the characteristic path length and clustering coefficient as a function of the length of the network. These are briefly discussed below for the three topological networks: regular, small world and random.

Scaling law 1 defines the behavior of the characteristic path length as a function of the network size, N. Intuitively one expects the characteristic path length to increase with the increase in the number of nodes in the network. It is observed that this increase is much faster for the regular network compared to the small world and random networks. In Figure 2.10, the characteristic path length as a function of the logarithm of the number of nodes, N, is plotted for regular (p = 0), small world (p = 0.06) and random (p = 1) networks in Figure 2.10 (a), 2.10 (b) and 2.10 (c) respectively. As regular networks are locally clustered, the characteristic path length increases linearly (exponentially on log scale) with the size of the network, as is evident in Figure 2.10 (a). In random and small world networks, the local clustering is disturbed by random rewiring of the edges, resulting in the reduction of the average path length. Thus, in these two cases, the average path exhibits a slower logarithmic increase with the size of the network (seen as a straight line on a semi-log plot).

Scaling law 2 defines the dependence of the clustering coefficient on the network size. Since from equation 2.5 it is evident that the clustering coefficient is dependent on the degree distribution of the network, the regular networks in which all the nodes have the same degree, the clustering coefficient is expected to be independent of the size of the network. In Figure 2.10 (d) for a regular network ($p = 0$), we do indeed observe that the clustering coefficient remains constant with increase in the number of nodes N. Small world networks exhibit strong local clustering similar to the regular networks, and so in this case also the clustering coefficient is found to be independent of the network size (shown in Figure 2.10 (e)). In the case of random networks, since nodes have a varying degree distribution, the clustering coefficient of the random graphs is given by ~ d/N, where d is the average degree and N is the size of the network suggesting a linear decrease with increase in network size. In Figure 2.10 (f) the clustering coefficient is indeed found to decrease with increasing N for the case with $p = 1$.

2.4 Protein contact networks are scale free

A systematic study was carried out by Greene and Higman (2003) on a set of 65 proteins and they showed that protein structures are scale free networks exhibiting small-world properties. They had considered protein structures as a system of interactions and analyzed the clustering coefficients and characteristic path lengths. In Figure 2.11, (reproduced from Greene and Higman, 2003) the clustering coefficient C and the characteristic path length L of the 65 representative proteins and the corresponding random networks are shown. The number of nodes and average connectivity is maintained the same in random networks as that in protein contact networks. Protein networks are shown in red and they cluster around C ~ 0.55, while random networks (in blue) cluster around C $\leq$ 0.1. From Figure 2.11 it is clear that the protein contact networks exhibit small world behavior, i.e.,

$$C_{protein\ networks} >> C_{random}$$

$$L_{protein\ networks} \sim L_{\ random}$$

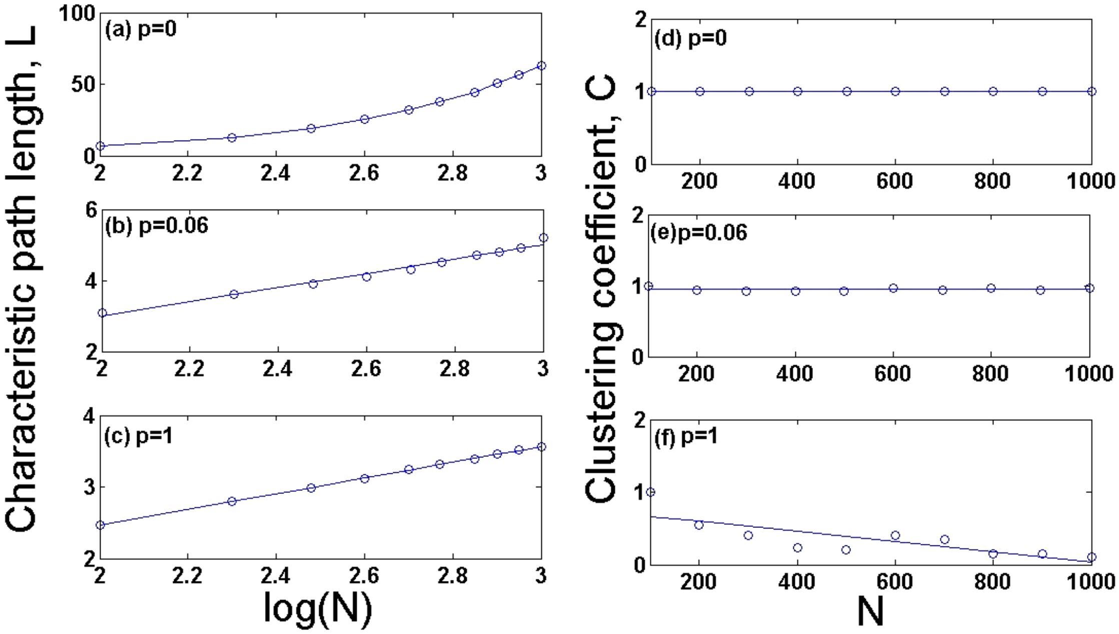


Figure 2.10: Characteristic path length and clustering coefficient as a function of network size, N, shown for various networks: (a), (d) Regular network, (b), (e) Small world network and (c), (f) Random network.

Figure 2.12 shows the plot of the fraction of nodes having degree *k*, *F(k) vs k,* where different colours correspond to proteins with 9 different folds (Greene and Higman, 2003). From this figure it is evident that for any set of similar colored dots, as the number of links increases, the number of nodes having those *k* links decreases following a power law (shown by a straight line fit). Since all the possible 9 folds are scale free, this implies that protein contact networks are scale free networks.

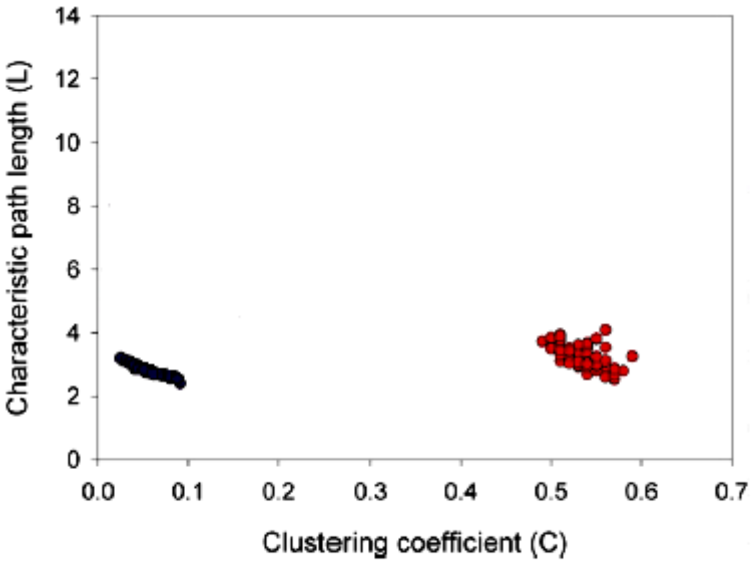


Figure 2.11: Small-world analysis of protein structures (Greene and Higman, 2003)

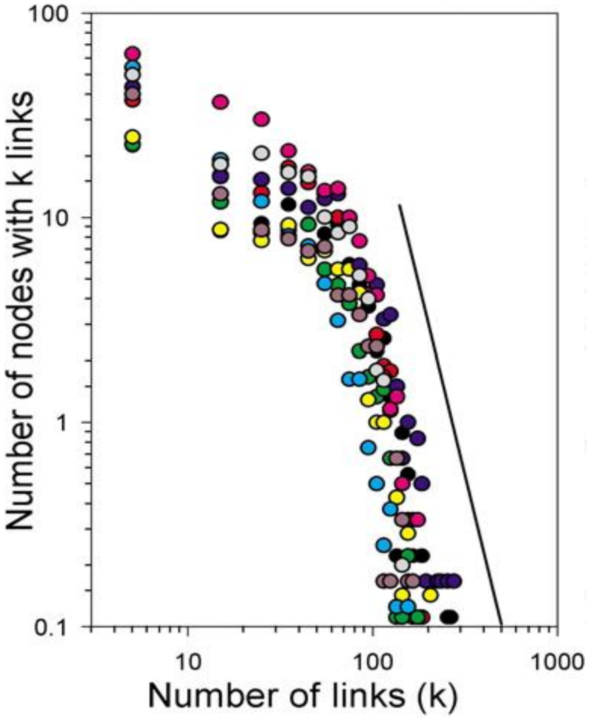


Figure 2.12: Plot of *F*(*k*) *vs.* *k* demonstrating scale free nature of the proteins (reproduced from Greene and Higman, 2003)

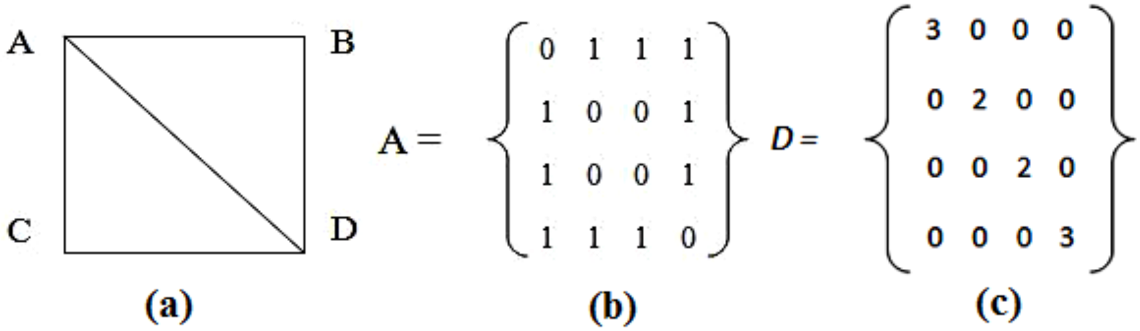
2.5 Mathematical representation of a graph

A graph can be represented as a mathematical entity by converting it into an algebraic form of a matrix which enables the detailed understanding of its connectivity. When represented as a matrix, analytical solutions of the graph can be obtained and various algorithms can be implemented to derive its properties. Many studies in graph theory like isomorphism, shortest path identification, spectral analysis, etc. have been performed on the matrices derived from the graph. The most commonly studied matrices associated with a graph are Adjacency and Laplacian matrices.

2.5.1 Adjacency matrix

Adjacency matrix of a graph with N nodes is an $N \times N$ symmetric matrix which contains the information about the connectivity of each node in the graph. The $(i, j)^{th}$ element, A_{ij} is either 1 or 0 depending on the presence or absence of a link between i and j as shown below in Figure 2.13(b), the adjacency matrix of the graph in Figure 2.13(a).

$$A_{ij} = \begin{cases} 1, \; if \;\; i \neq j \;\; and \;\; edge \; between \, i \;\; and \;\; j \\ 0, \; if \;\; i = j \;\; or \;\; no \;\; edge \; between \, i \;\; and \;\; j \end{cases} \tag{2.8}$$



$$A = \begin{Bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{Bmatrix} \quad D = \begin{Bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{Bmatrix}$$

(a) **(b)** **(c)**

Figure 2.13: (a) Sample graph, (b), (c) Adjacency matrix and Degree matrix of the graph in (a).

2.5.2 Degree Matrix

The degree matrix is a diagonal matrix which gives the information about the degree of each vertex in the graph (Figure 2.13(c)). It is obtained by summing up the columns or rows in the adjacency matrix.

$$D_{i\,j} = \begin{cases} \deg(V_i) \ \mathit{if} \ i = j \\ 0 \ \mathit{otherwise} \end{cases} \tag{2.9}$$

2.5.3 Laplacian Matrix

The Laplacian matrix L of a graph is defined as: $L = D - A$ where D is the degree matrix and A the adjacency matrix of the graph. The Laplacian matrix corresponding to the graph in Figure 2.13(a) is:

$$L = \left\{ \begin{matrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{matrix} \right\}$$

2.6 Graph Spectra

The set of eigenvalues and eigenvectors of the adjacency or Laplacian matrix is called the spectrum of the graph. Graph spectral theory is concerned with the relationships between the algebraic properties of the spectra of these matrices associated with the graph and the topological properties of the graph. They provide information on the structure and topology of the graph and analysis of these quantities is known as *graph spectral analysis* (Vishveshwara, 2002). Spectral techniques are commonly used in the design of circuits, VLSI chips and computer networks. Identification of clusters and similarity in connectivity patterns can be deduced from the spectral analysis of connected graphs

(Gould, 1967). The graph spectral analysis has yielded valuable results in the identification of clusters in protein structures (Kannan and Vishveshwara, 1999; Patra and Vishveshwara, 2000).

An eigenvalue of a square matrix *A*, represented as λ, is defined as a scalar quantity satisfying the equation $\boldsymbol{Ax = \lambda x}$ if there exists a non-zero vector *x*, called an eigenvector (corresponding to λ), and the pair (λ, *x*) is called an eigenpair for *A*. The k^{th} principal eigenvector of a graph is defined as the eigenvector corresponding to the k^{th} largest eigenvalue of *A*. The eigenvector corresponding to the largest eigenvalue is referred to as the principal eigenvector. It provides a measure for the centrality of the vertices in the graph.

The eigenvalues and vector components of the adjacency matrix of the graph in Figure 2.13(a) are given in Table 2.1(a) and for Laplacian matrix in Table 2.1(b) respectively. The vector components corresponding to the largest eigenvalue of the adjacency matrix, 2.56 in this case, provide information regarding the contribution of each node in the graph, i.e., the number of other nodes connected to it. A highly branched node connected to other highly branched node is identified as one with the largest absolute vector component value. In graph 2.13(a), the degree of the nodes A and D is 3, while that of nodes B and C is 2. The magnitude of the vector components of the largest eigenvalue of both the adjacency and Laplacian matrices tabulated in the Table 2.1(a) and 2.1(b) respectively reflects this observation. The eigenvectors of the leading eigenvalue corresponding to nodes A and D have same value, similarly for nodes B and C as these pairs have same connectivity pattern. The largest eigenvalue and its corresponding eigenvector of adjacency matrix have been shown to be useful in the identification of charged, hydrophobic and backbone clusters (Sistla *et al*, 2005). In this study we show that the spectral analysis of these matrices can help in identifying compact structural modules, domains.

Table 2.1(a): Eigenvalues and eigenvectors of adjacency matrix of the graph in Figure 2.13(a)

S.no.	Eigenvalue	Eigenvector
1	2.56	0.55, 0.43, 0.43, 0.55
2	5.77 e-32	1.07e-16, 0.707, -0.707, 5.66e-17
3	-1.00	0.70, 1.78e-16, 3.45e-16, -0.70
4	-1.56	-0.43, 0.55, 0.55, -0.43

Table 2.1(b): Eigenvalues and eigenvectors of the Laplacian matrix of the graph in Figure 2.13(a)

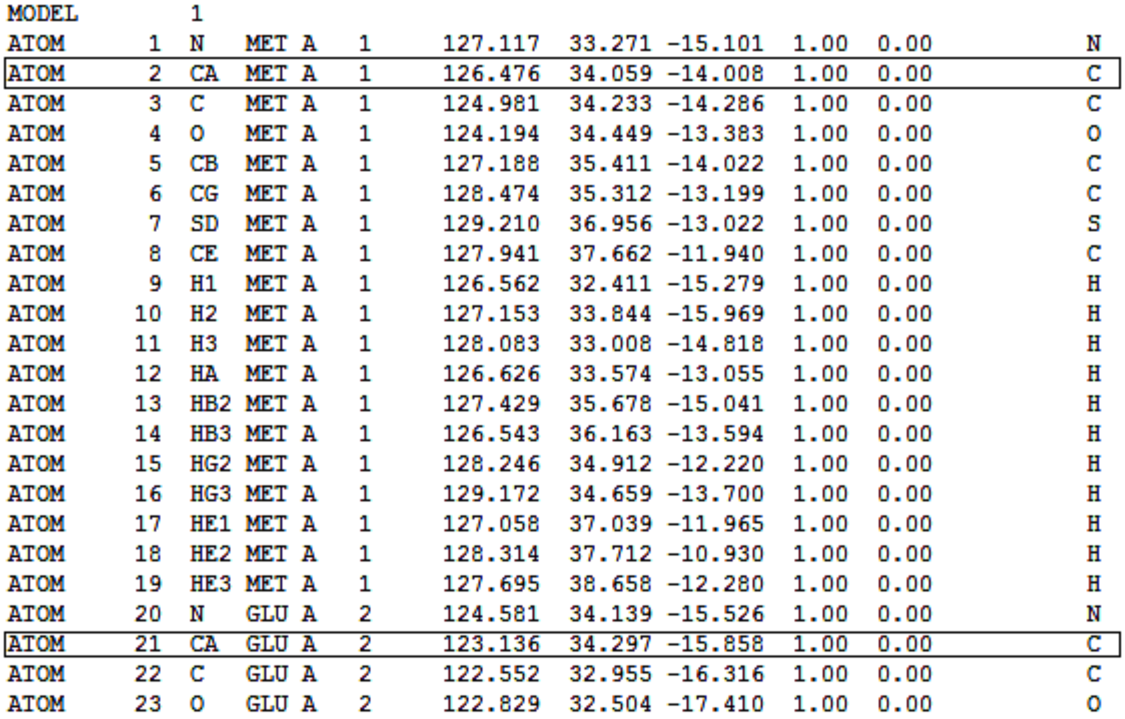
S.no.	Eigenvalue	Eigenvector
1	4	-0.81, 0.408, 0.408, -0.81
2	4	0.86, -0.28, -0.28, 0.86
3	2	0, -0.707, 0.707, 0
4	1.11e-16	0.5, 0.25, 0.25, 0.5

2.7 Constructing Protein Contact Graphs

Various approaches have been proposed for the construction of protein contact networks in an attempt to capture the topology of the protein structure. These vary from graphs at a coarse-grained level mimicking interactions between secondary structure elements, to that between backbone carbon atoms, C_α, or, at a more finer level, capturing interactions between side-chain C_β atoms, or, between all the atoms of the graph, depending on the questions asked and the analysis to be performed. Koch and co-workers (Koch *et al*, 1992) introduced the concept of a mathematical graph to represent β structures. They represented a single β strand as a vertex and two separate edge sets describing the sequential and hydrogen bond connections respectively. Later Grigoriev and coworkers (Grigoriev *et al*, 1994) represented all α helical structures in the form of connected graphs in which nodes represent helical secondary structures and the edges represent contacts between helices, rather than hydrogen bonds which was the case with β graphs. Secondary structure graphs (α graphs and β graphs) have been applied for protein fold recognition (Mitchell et al, 1989). Nishikawa *et al* (1972) proposed the maps constructed using the backbone C_α atoms. Vishveshwara *et al* (1999) constructed protein graphs using C_β atoms (the first carbon atom belonging to the side chain and directly attached to C_α) as nodes for the identification of side-chain clusters in protein structures. All atom graphs, obtained by considering each atom as a node was given by Jacobs *et al* (2001) for predicting protein flexibility. Thus different representations of the protein graphs have been proposed to serve different purposes. In this study the protein contact networks have been constructed by considering the backbone C_α atoms as nodes with an edge drawn between two C_α atoms if they are either connected by a peptide bond or lie within a cut-off distance, R_c. The value of R_c is chosen as an upper limit to the range of non-covalent interactions that are known to play a significant role in the three dimensional fold of the protein (Sistla *et al,* 2005). The Euclidian distance between two C_α atoms is computed as follows:

$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \qquad (2.10)$$

by extracting the 3-dimensional coordinates of the atoms of a protein molecule from the record file in PDB (Berman *et al*, 2000). A snapshot of the relevant information in a PDB record containing coordinates of all its atoms is shown in Figure 2.14, with the coordinates of the C_α atoms highlighted in a rectangular box. The construction of the protein network is illustrated in Figure 2.15 where an edge is drawn from the *i*th node to all other nodes lying within a circle of radius 7Å centered at the *i*th node, i.e., to the nodes *i*-1, *i*+1, *i*+2, A, B, C and D. Nodes A, B, C, D are called spatial neighbors of *i* (connected by dotted lines) while nodes *i*-1, *i*+1, *i*+2 are called the sequential neighbors of *i* (connected by continuous lines).



```
MODEL        1
ATOM      1  N   MET A   1     127.117  33.271 -15.101  1.00  0.00           N
ATOM      2  CA  MET A   1     126.476  34.059 -14.008  1.00  0.00           C
ATOM      3  C   MET A   1     124.981  34.233 -14.286  1.00  0.00           C
ATOM      4  O   MET A   1     124.194  34.449 -13.383  1.00  0.00           O
ATOM      5  CB  MET A   1     127.188  35.411 -14.022  1.00  0.00           C
ATOM      6  CG  MET A   1     128.474  35.312 -13.199  1.00  0.00           C
ATOM      7  SD  MET A   1     129.210  36.956 -13.022  1.00  0.00           S
ATOM      8  CE  MET A   1     127.941  37.662 -11.940  1.00  0.00           C
ATOM      9  H1  MET A   1     126.562  32.411 -15.279  1.00  0.00           H
ATOM     10  H2  MET A   1     127.153  33.844 -15.969  1.00  0.00           H
ATOM     11  H3  MET A   1     128.083  33.008 -14.818  1.00  0.00           H
ATOM     12  HA  MET A   1     126.626  33.574 -13.055  1.00  0.00           H
ATOM     13  HB2 MET A   1     127.429  35.678 -15.041  1.00  0.00           H
ATOM     14  HB3 MET A   1     126.543  36.163 -13.594  1.00  0.00           H
ATOM     15  HG2 MET A   1     128.246  34.912 -12.220  1.00  0.00           H
ATOM     16  HG3 MET A   1     129.172  34.659 -13.700  1.00  0.00           H
ATOM     17  HE1 MET A   1     127.058  37.039 -11.965  1.00  0.00           H
ATOM     18  HE2 MET A   1     128.314  37.712 -10.930  1.00  0.00           H
ATOM     19  HE3 MET A   1     127.695  38.658 -12.280  1.00  0.00           H
ATOM     20  N   GLU A   2     124.581  34.139 -15.526  1.00  0.00           N
ATOM     21  CA  GLU A   2     123.136  34.297 -15.858  1.00  0.00           C
ATOM     22  C   GLU A   2     122.552  32.955 -16.316  1.00  0.00           C
ATOM     23  O   GLU A   2     122.829  32.504 -17.410  1.00  0.00           O
```

Figure 2.14: A portion the PDB file of protein1A5E showing the coordinates of its atoms.

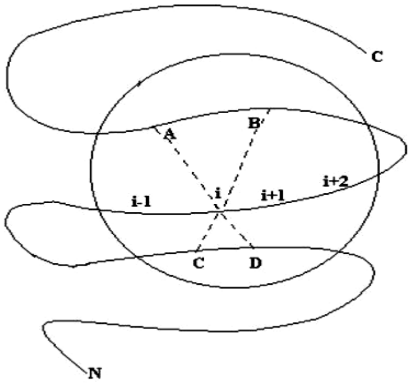


Figure 2.15: A neighbours of the i^{th} node shown (reproduced from Vishveshwara *et al,* 2002).

Thus in this case the adjacency matrix is defined as:

$$A_{ij} = 1, \; if \;\; d_{ij} \leq R_c \;\; and \; i \neq j \qquad (2.11)$$

$$A_{ij} = 0, \; if \;\; d_{ij} > R_c \;\; and \; i = j$$

where the Euclidean distance, d_{ij}, between every (i, j) pair is 7Å.

2.8 Domain Identification

Proteins are intrinsically modular, i.e., they are assembled from smaller structural modules that are spatially separable and can fold into independent compact 3-dimensional shapes known as *domains*. Genetic recombinant techniques allow reorganization of domains at the amino acid sequence level (such as circular permutations). This along with domain swapping, domain insertion and domain accretion (discussed in Chapter 1, section 1.2) play a major role in the functional evolution of proteins. The structural classification of proteins in databases such as SCOP and CATH is at the domain level as domains can fold independently and form distinct structural and functional units. The decomposition of proteins into domains also reduces the complexity in analyzing the function of proteins. Also, the methods for phylogenetic analysis and protein modeling usually work best for single domains (Ponting and Russell 2002). Hence there clearly exists a need for accurate determination of the domains. Thus understanding the evolution of domains is important in understanding the functionality of the proteins. The existence of non-contiguous domains adds to the difficulties in developing an automated solution for domain partitioning.

The identification of structural domains in proteins is based on the assumption that the interactions between the amino acids are higher within a domain than across the domains. These interactions and the topological details of protein structures can be effectively captured by considering protein structure as a graph. The densely connected groups of vertices in such a graph would then correspond to compact structural domains. Detection of densely connected clusters has received considerable attention because of their significant practical importance in a number of real-world networks. For instance, groups within the worldwide web might correspond to sets of web pages on related topics (Flake *et al*, 2002); groups within social networks might correspond to related social units or communities (Girvan and Newman, 2002). In metabolic networks these could provide evidence for a modular view of the network's dynamics, with different groups of nodes performing different functions with some degree of independence. Two different approaches used in the detection of groups in networks are graph partitioning and hierarchical clustering.

In the first approach, given a graph $G = (V, E)$, where V is the set of vertex and E the set of edges that determines the connectivity between the nodes, the graph partitioning problem consists of dividing a graph G into n disjoint partitions. This is a top-down approach and involves splitting the graph into predefined groups by minimizing the number of edge cuts between groups. This approach has wide applications in community detection in WWW, image segmentation, partitioning the data and assigning tasks among the processors of a parallel computer, etc. (Elsner, 1997; Fjallstrom, 1998). A number of algorithms have been proposed for graph partitioning. The most commonly used is Kernighan–Lin algorithm (Kernighan and Lin, 1970). For partitioning a graph into two disjoint subsets A and B, the algorithm attempts to find an optimal series of interchange operations between the partitions A and B which minimizes the number of edge cuts between nodes in A and B. Graph partition problem is proved to be NP hard (Garey *et al,* 1976), thus it is unlikely that there is a polynomial time algorithm that always finds an optimal solution. The algorithm by Karisch *et al* (1997) can be used on graphs with less than 100 nodes but are too slow on larger graphs. These algorithms works best if we know beforehand the number and size of the groups into which the network is to be split. Also, the goal is to find best division of the network regardless of whether a good division even exists or not. Therefore, all partition algorithms are heuristics and differ with respect to cost (time and memory space required to run the algorithm) and partition quality (Per-Olof, 1998).

The second approach, hierarchical clustering has been pursued by sociologists and by physicists and biologists, with applications particularly in social and biological networks (White *et al*, 1976; Wasserman and Faust, 1994; Newman, 2004). In this approach, a hierarchy of clusters is built and it is generally classified into two types- agglomerative (bottom-up) and divisive (top-down). In order to decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required. The applications of hierarchical clustering are immense in various fields like biology, medicine, and market research; extending into social network analysis, image segmentation, data mining, search result grouping, mathematical chemistry, etc. Hierarchical clustering in contrast to graph partitioning is best suitable for networks like WWW, social network, biological networks, etc. as these methods assume that the network divides naturally into subgroups. Thus the number and size of the groups are determined by the network itself, no prior information on the number and size of the groups is required. Moreover, hierarchical methods will explicitly indicate that no good division of the network exists, if such be the case (Newman, 2006). Despite its utility, hierarchical clustering has many flaws. Interpretation of the hierarchy is complex and often confusing; the deterministic nature of the technique prevents reevaluation after points are grouped into a cluster; all determinations are strictly based on local decisions and a single pass of analysis and the fact that the intra cluster distance may be different for different clusters gets ignored here. Using hierarchical clustering, domains can be identified in two ways: (i) Top-down approach, i.e., starting with the whole structure, we proceed by partitioning it iteratively into smaller and compact structural units, and (ii) Bottom-up approach, i.e., start by defining very small structural units, e.g. secondary structural elements and assembling them into domains. Here, we use a combination of both these approaches: we first decompose the protein contact network into compact structural modules and then assemble them to predict true domains (Hari Krishna and Parekh, 2009). The details of the proposed approach for domain identification using hierarchical clustering are given below.

In this study we have implemented the modularity based graph spectral approach proposed by Newman (Newman, 2006) for identifying domains in the three-dimensional structure of proteins. Newman's approach has been originally proposed for the detection and characterization of community structure in social networks, i.e., the appearance of densely connected groups of vertices, with only sparser connections between the groups. A similar situation is observed in multi-domain proteins where each domain forms a compact structural module with amino acids within a domain are in close proximity with each other than with the amino acids of other domains. In other words, the number of edges running between two domains (inter-domains) is much fewer than those within domains (intra-domains). The strong intra-domain interaction can be seen in the C_α-C_α distance plot as blue patches (in Figure 2.16 from residues 1-190 and 210-400) for a 2-domain protein *Phosphoglycerate kinase* (PDB ID: 16PK(A)). Since theoretically the definition of a community structure (in social networks) and domains (in protein structures) are similar, we expect Newman's community detection algorithm to accurately identify structural domains in proteins. Below we briefly discuss the Newman's approach and its implementation details on protein contact networks.

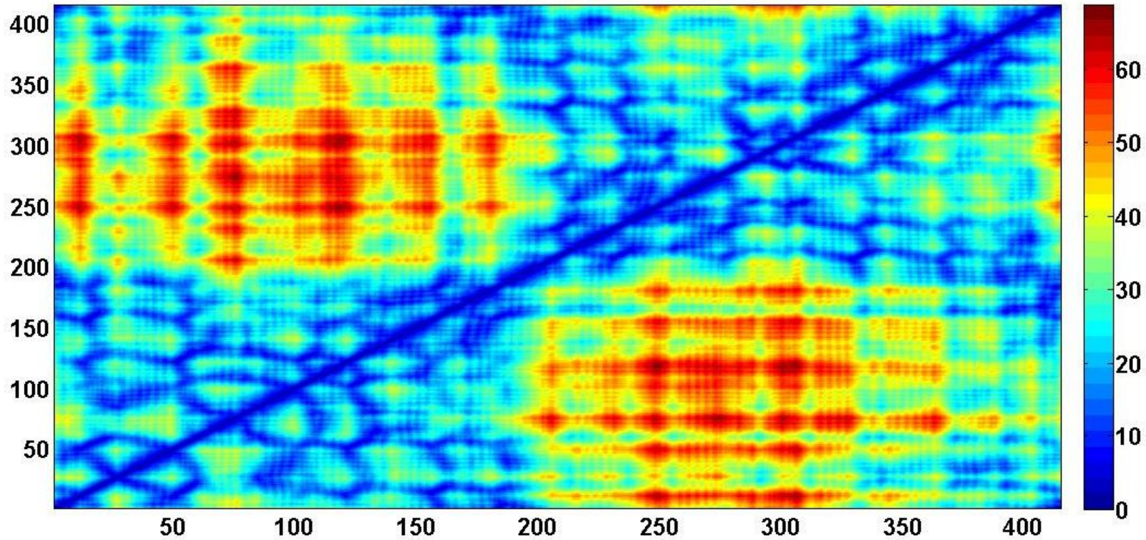


Figure 2.16: Contour plot of the Cα-Cα distance matrix of the protein 16PK(A)

2.8.1 Newman's Modularity Approach

In 2006 Newman proposed a modularity-based approach for identifying community structure in social networks. That is, given a network, to determine whether there exists any natural division of its vertices into non-overlapping groups or communities (of any size). The approach looks for the most natural subdivisions of the network rather than partitioning a network into pre-defined groups. The basic principle of this approach is to look for divisions of the vertices into two groups so as to minimize the number of edges running between the groups. It is proven that maximizing the modularity is the preferred approach in community detection and works well for most of the real world networks (Guimera and Amaral, 2006; Leon Danon *et al,* 2005). This approach is most suitable for domain identification problem as one may not have *a priori* knowledge of the number of structural domains in a protein.

The first step of the approach is to look for the possibility of the division of the network into just two groups, i.e., to look for division of the vertices into two groups so as to minimize the number of edges running between the groups. This ''minimum cut'' approach is commonly adopted in the graph-partitioning literature. According to Newman, a good division of a network into groups is not merely one in which there are few edges between groups but the one with *fewer than expected* edges between groups. Hence to assess the quality of the division, Newman has defined a measure called *modularity*, which is the difference in the number of edges falling within groups and the expected number in an equivalent network with edges placed at random. In our domain identification problem, this would amount to identifying structural groups having fewer inter-domain connections than expected by random chance. The modularity can be either positive or negative, with positive values indicating the possible presence of modular structure.

For the initial division of the protein contact network with N vertices into two groups, let $s_i = 1$ if vertex i belongs to group 1 and $s_i = -1$ if it belongs to group 2. The adjacency matrix A for a protein contact network is defined as:

$$A_{ij} = 1,\ if\ \ d_{ij}\ \le R_c\ \ and\ i \ne\ j \tag{2.12}$$

$$A_{ij} = 0,\ if\ \ d_{ij}\ > R_c\ \ and\ i\ =\ j$$

where d_{ij} is the Euclidean distance between every (i, j) node pair and an edge is drawn between the nodes if $d_{ij} \le R_c$ (7Å). The number of edges between vertices i and j is given by the elements A_{ij} of the adjacency matrix, which take the value '1' or '0' depending on whether two residues are within a distance of R_c of each other or not. The expected number of edges P_{ij} between vertices i and j if edges are placed at random is given by

$$P_{ij} = \frac{k_i k_j}{2m} \tag{2.13}$$

where k_i and k_j are the degrees of the vertices i and j respectively, and m is the total number of edges in the network, i.e.,

$$\boldsymbol{m = \frac{1}{2}\sum_i k_i} \tag{2.14}$$

Modularity, Q, is then defined as the sum of $A_{ij} - P_{ij}$ computed over all pairs of vertices (i, j) falling in the same group, i.e.,

$$Q = \frac{1}{4m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\left(s_i s_j + 1\right) \tag{2.15}$$

$$= \frac{1}{4m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\left(s_i, s_j\right)$$

where $s_i = 1$ if vertex i belongs to group 1 and $s_i = -1$ if it belongs to group 2, so that the quantity $1/2(s_i s_j+1)$ is 1 if i and j are in the same group and 0 otherwise. Here, A_{ij} and P_{ij} are the observed and expected number of edges and m the total number of edges in the network. This equation can conveniently be written in a matrix form as:

$$Q = \frac{1}{4m} s^T Bs \tag{2.16}$$

where s is the column vector whose elements are s_i and B is a real symmetric matrix with elements:

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m} \tag{2.17}$$

and is called the modularity matrix. The spectral properties of this modularity matrix, B, are analyzed for identifying clusters.

The leading eigenvector of the modularity matrix, B, is computed and the vertices divided into two groups according to the sign of the elements in this vector, i.e., all vertices whose corresponding elements are positive are placed in one group and the rest in the other group. This is depicted in Figure 2.17, wherein the eigenvector of the largest eigenvalue of the modularity matrix is plotted for the protein graph 1SW6 (chain A). Thus in this case the network will be split into two subgroups: 1 to 156 and 157 to 286 based on the sign of the eigenvector components of the largest eigenvalue. If there are no positive eigenvalues of the modularity matrix, the leading eigenvector would be the vector (1,1,1, . . .), i.e., all eigenvector components would be of the same sign. In this case the algorithm indicates that there is no division of the network possible and all vertices are put in a single group. This is an important feature of the algorithm since it suggests that the algorithm has the ability not only to divide networks effectively, but also to refuse to divide them when no good division exists. Thus, a network is indivisible if the modularity matrix has no positive eigenvalues. We thought that this would be a very useful feature in the domain identification of proteins since a large number of proteins are single domain proteins. Though the algorithm makes use only of the signs of the elements of the leading eigenvector, but the magnitudes too convey information. Vertices corresponding to elements of large magnitude make large contributions to the modularity, and conversely for small ones. For protein contact network we expect this to give useful information regarding residues important for the fold and stability of the domain.

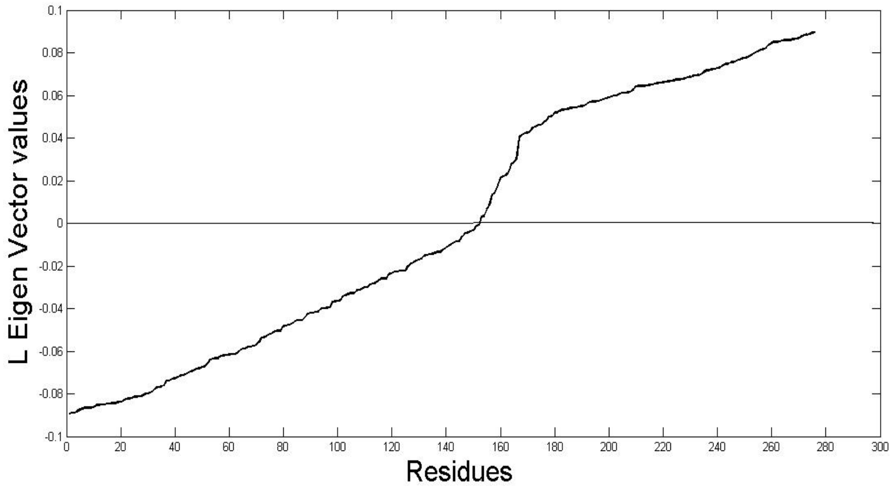


Figure 2.17: Principal eigenvector plot for the Modularity matrix of 1SW6 (chain A) with residue numbers on X axis and eigenvector values on Y axis.

Since many networks may contain more than two groups, for e.g., a protein structure may contain 2 or more domains, the method should identify more than 2 divisions also. The standard approach to this problem is repeated division into two, i.e., using the above algorithm first to divide the network into two parts then divide each of the parts into two and so on. When splitting a network beyond two groups, the degrees k_i and k_j for the calculation of the modularity Q will change if edges falling between the two groups are deleted. Thus instead of calculating absolute modularity Q at each split, an additional contribution ΔQ to the modularity upon further dividing a group (cluster) g of size n_g in two is computed as follows (Newman, 2006):

$$\Delta Q = \frac{1}{4m} \sum_{i,j \in g} \left[B_{ij} - \delta_{ij} \sum_{k \in g} B_{ik} \right] s_i s_j$$

$$= \frac{1}{4m} s^T B^{(g)} s \qquad (2.18)$$

where δ_{ij} is the Kronecker δ-symbol, and $B^{(g)}$ is the $n_g \times n_g$ matrix with elements indexed by the labels i, j of vertices within group g and having values

$$\boldsymbol{B}_{ij}^{(g)} = \boldsymbol{B}_{ij} - \boldsymbol{\delta}_{ij} \sum_{k \in g} \boldsymbol{B}_{ik} \qquad (2.19)$$

Since equation 2.18 has the same form as equation 2.16, the spectral approach can be applied to this generalized modularity matrix also, to maximize ΔQ.

A nice feature of this method is that, in repeatedly subdividing the network it addresses the question of when to stop the subdivision process. If there exists no division of a subgraph that will increase the modularity of the network, or equivalently that gives a positive value for ΔQ, then there is nothing to be gained by dividing the subgraph and it is left undivided. This happens when there are no positive eigenvalues to the matrix $B^{(g)}$. Thus the leading eigenvalue provides a simple check for the termination of the subdivision process: if the leading eigenvalue is zero, which is the smallest value it can take, then the subgraph is indivisible.

According to Newman, the subdivision with the largest value of modularity is considered as the best natural subdivision of the network. When Newman's approach was implemented on the protein contact network of a multi-domain protein, the network was split into a number of spatially compact structural modules or motifs of varying sizes; some of them as small as 10-20 residues. In many cases the clusters formed were comprised of short non-contiguous peptide segments. That is, the algorithm did not stop the decomposition process on identifying the domains but continued the process resulting in much smaller closely packed secondary structure elements. These structural units actually correspond to best subdivisions on the basis of modularity value. Thus, though one of the intermediate subdivisions of the network did correspond to the true domains, it did not have the highest modularity value making it difficult to decide when to stop the division of the network for the domain identification problem. Hence, we propose a bottom-up approach of clustering the compact structural modules to predict true domains. This was done by first constructing a coarse protein structure graph on treating each cluster (from Newman's approach) as node and these clusters are connected with weighted edges, the weights being proportional to the interaction strength, i.e., number of (x_i, y_j) pair of residues that lie within 7Å distance, where x_i is a node belonging to cluster X and y_j is a node belonging to cluster Y. The weights are computed by the formula

$$e_{XY} = \left[\frac{N_{XY}}{(N_X + N_Y)} \right] \times 100 \tag{2.20}$$

where N_{XY}corresponds to the total number of interactions between the residues of clusters X and Y, normalized by the total size of the two clusters, N_X and N_Y. A representative coarse-network is shown in Figure 2.18 where the solid color filled circles correspond to the initial nodes of the protein contact network which are grouped into four clusters represented by dotted circles. Now each dotted circle is considered as a node and a reduced graph consisting of four nodes A, B, C, D is obtained with weighted edges (Figure 2.18(b)). Two pairs of nodes between clusters C and D are interacting (within 7Å), shown as solid lines between C and D, so $N_{CD} = 2$. Since the number of nodes in cluster C, $N_C = 5$ and in cluster D, $N_D = 4$, the weight of the edge connecting C and D, $e_{CD} = (2/(5+4)) * 100 = 22.22$. Similarly the weights of all the other edges are computed. We next implement the agglomerative approach by Clauset *et al* (2004) on this weighted protein structure graph to identify true structural domains. The approach is briefly explained below.

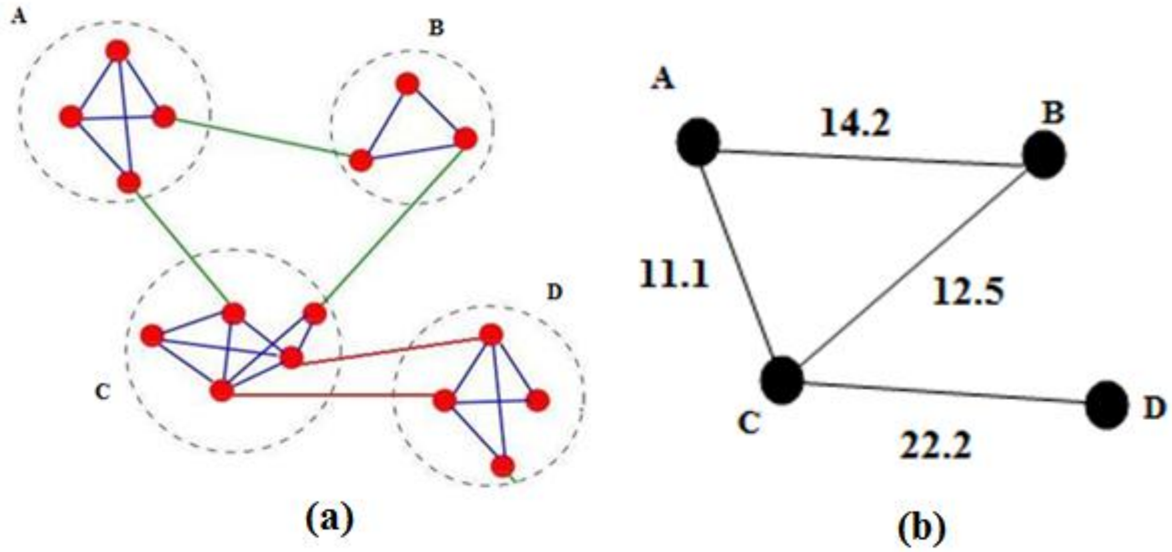


Figure 2.18: (a) The solid color filled circles are the initial nodes of the network which are clustered into 4 groups by Newman's approach, represented by dotted circles. Now each dotted circle is considered as a node for the construction of coarse protein structure graph. (b) Reduced graph of the Figure 2.18(a).

2.8.2 Agglomeration of compact structural modules into domains

The approach used by Clauset *et al* is a bottom-up hierarchical agglomerative approach, i.e., it starts with treating each of the *N* nodes as clusters and agglomerates them by a greedy optimization of modularity, i.e., for a graph of *N* nodes there will be *N*-1 merges, with the *N*-1th merge corresponding to all nodes in a single cluster and having the highest modularity value, 1.

An adjacency matrix providing the connectivity information of this coarse protein structure graph is given by

$$A_{vw} = \begin{cases} 1, \; if \;\; nodes \; v \; and \;\; w \; are \; connected \\ 0 \; otherwise \end{cases} \tag{2.21}$$

where the node v belongs cluster C_v and node w to cluster C_w respectively. Then the fraction of edges that fall within these clusters *i.e.,* connecting vertices lie in the same cluster or group:

$$\frac{\sum_{vw} A_{vw}\delta(C_v, C_w)}{\sum_{vw} A_{vw}} = \frac{1}{2m}\sum_{vw} A_{vw}\delta(C_v, C_w) \tag{2.22}$$

where the δ-function, $\delta(i,j) = 1$, if $i = j$ and 0 otherwise, and $m = ½ \, \Sigma_{vw}A_{vw}$ is the number of edges in the graph. This quantity will be large for good divisions of the network, in the sense of having many within-group edges, but is not a good measure since it takes its largest value of 1 in the trivial case of all vertices belonging to a single group. However, as before, if we subtract from it the expected value of the same quantity in the case of a randomized network, i.e., $k_vk_w/2m$, then a modularity measure is again defined to assess the amalgamation at each step:

$$Q = \frac{1}{2m}\sum_{vw}\left[A_{vw} - \frac{k_v k_w}{2m}\right]\delta(C_v, C_w) \tag{2.23}$$

When the greedy agglomerative algorithm is implemented on the coarse protein structure graph (Figure 2.12(b)), the clusters are repeatedly merged and the process is continued till the amalgamation produces the largest modularity Q. The merging is done based on the weights of the edges, i.e., the pair of clusters with highest edge weight between them is merged first and so on. The resultant clusters with largest value Q are then reported as true domains. This two-phase (top-down & bottom-up) approach of domain identification is particularly useful when the domains are non-contiguous along the polypeptide chain as shown in Figure 2.19. For protein 1HLE(A) domain I (enclosed by pink rings) comprises of two discontinuous regions (1-27 and 171-276) and Domain II (enclosed by blue rings), is also formed by two discontinuous regions (27-170 and 277-341) as shown in Figure 2.19.

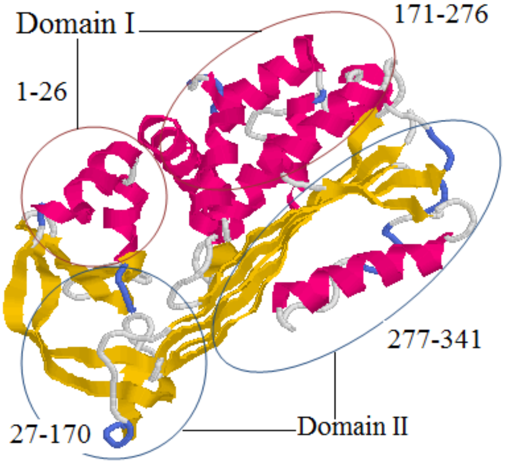


Figure 2.19: The 3-dimensional structure of 1HLE(A).

The proposed method works well for multi-domain proteins (Chapter-3). However for the single domain proteins this approach fails. The first phase (top-down approach) divides the network into compact secondary structures. When the second phase, bottom-up approach is implemented the highest modularity amalgamation step always being one, corresponding to a single group, it is not possible to identify whether a given protein consists of single or multiple domains. Hence before implementing amalgamation approach, we need to know whether the protein is a single domain or a multi domain protein. To see if any of the graph properties would help in differentiating between single and multi-domain proteins, we analyzed the following properties: Diameter, Density, Clustering coefficient and Split-1 interactions in the first phase of our top-down approach. Of all these measures, we observe that density and split-1 interaction strength are both useful in differentiating between single domain and multi-domain proteins. The number of interactions between the two subgroups obtained from the first subdivision of the network by Newman's algorithm (Newman, 2006) is computed and referred here as split-1 interaction strength (IS-1). This corresponds to the number of residue pairs between the two substructures falling in the range of $7A^o$ distance. The length of the protein is also important in this regard since typically single domain proteins are of length < 300 (ref). The results of our analysis are discussed in Chapter 3.

2.9 Implementation details

A Perl script was written to parse the coordinates of the protein structure from the PDB file and compute the distance matrix. Using this distance matrix, all C_α atom pairs within 7Å are connected by an edge and this information is stored in an "edge list", listing pairs of nodes between which an edge exists. This edge list file forms input to the Newman's approach to compute the modularity matrix and the eigenvector components of its leading eigenvalue. The clusters of strongly connected compact structural modules then form input to the Clauset *et al* algorithm to predict the true structural domains. Implementation of partitioning and amalgamation phases is done in C++ using igraph library (Csardi and Nepusz, 2006).

2.9.1 Libraries and Tools

Implementation of Newman's algorithm and Clauset's fast greedy modularity optimization algorithm is done in C++ using the igraph library, which is an open source and distributed under the terms of the GNU GPL for creating and manipulating graphs. The Protein Data Bank (PDB) is used for obtaining the structural coordinates of the C_α atoms. MATLAB is used for drawing protein contact networks by using the atomic coordinates in the PDB file. RASMOL is used for the three dimensional visualization of the proteins.

2.9.2 Data Set

Analysis has been carried out on a set of 100 proteins of which 50 are single and 50 are multi-domain proteins. About 23 of the multi-domain proteins have non-contiguous domains along the polypeptide chain. These are listed in Table 2.2 and belong to different structural classes, *viz*., α, β, α/β and α+β. The Jones *et al* (1998) dataset consisting of 55 proteins (21 multi-domain and 34 single domains) have been used to evaluate the performance of our approach with other structure based domain prediction methods such as DETECTIVE (Swindells, 1995), PUU (Holm and Sander, 1994), DOMAK (Siddiqui and Barton,1995), DomainParser (Xu *et al,* 2000). The results of 100 proteins are compared with the annotations in CATH database (Orengo *et al,* 1997) and with the results of DomainParser.

Table 2.2: Dataset of 100 proteins used for the analysis

S. No.	PDB ID	Structural Class	Resolution (Å)	Length	Average Path length	Clustering Coefficient
1	16PK (A)	α/β	1.6	415	6.94942	0.492942
2	1ATN (A)	α+β	2.8	373	6.63051	0.511267
3	1BBH (A)	α	1.8	131	4.71638	0.538248
4	1BBP (A)	Mainly β	2.0	173	5.2382	0.533073
5	1BKS (A)	α/β	2.2	268	6.90119	0.54357
6	1BKS (B)	α/β	2.2	397	5.86535	0.495969
7	1BRD	α	3.405e+38	248	5.18622	0.534821
8	1CDG (A)	Mainly β	2.0	686	6.54635	0.496012
9	1D5M (B)	α+β	2.0	192	4.50522	0.505141
10	1EPW	β	1.9	1290	8.70291	0.539807
11	1EZM (A)	α+β	1.5	301	6.81206	0.536001
12	1FNB (A)	β	1.7	314	6.83408	0.538104
13	1FXI	α+β	2.2	96	4.3057	0.544003
14	1G6N (A)	A	2.1	210	5.59025	0.536669
15	1GKY	α/β	2.0	187	5.26806	0.530284
16	1GMP(A)	α+β	1.7	96	4.3057	0.544003
17	1GOX	α/β	2.0	370	6.49618	0.51314

18	1GPB	α/β	1.9	842	7.44441	0.571058
19	1GPD (A)	α/β	2.9	334	6.67464	0.523757
20	1HLE (A)	α/β	1.95	345	8.67918	0.505769
21	1KF6 (B)	α/β	2.7	243	6.73673	0.514333
22	1LAM (A)	α/β	1.6	484	8.40434	0.591391
23	1LAP (A)	α/β	2.7	487	6.47	0.518222
24	1LFI (A)	α/β	2.1	691	6.73719	0.509949
25	1MSH (A)	α+β	NMR	72	4.94405	0.577653
26	1OFV	α/β	1.7	169	5.19463	0.536394
27	1PFK (A)	α/β	2.4	320	6.67866	0.529954
28	1PGR (B)	β	3.5	215	4.94631	0.505445
29	1PHA (A)	α	1.6	414	6.60116	0.519858
30	1PHH (A)	α/β	2.3	394	6.61659	0.517155
31	1PPN	α+β	1.6	212	5.79867	0.538549
32	1PPR (M)	α	2.0	312	6.63051	0.511267
33	1PRC (C)	α	2.3	336	6.76933	0.541448
34	1PYP	β	3	285	6.76933	0.541448
35	1RBP	β	2.0	182	5.25912	0.532584
36	1RCB	α	2.2	129	4.70821	0.541237
37	1RHD (A)	α/β	2.5	293	6.82199	0.537931
38	1RVE	α/β	2.5	245	7.18417	0.544391
39	1SGT (A)	β	1.7	223	6.09221	0.540152
40	1SMP (A)	β	2.3	471	8.47659	0.58475
41	1SMR	β	2.0	335	8.47659	0.58475
42	1SNC	β	1.6	149	4.72527	0.536278
43	1SU4	α+β	2.4	994	7.42738	0.549449
44	1TAH	α/β	3.0	318	7.19345	0.626127
45	1TIE	β	2.5	172	5.12406	0.534904

46	1TLK	β	2.8	154	4.36493	0.536748
47	1UGO (A)	α	NMR	99	5.20903	0.570879
48	1ULA	α/β	2.7	289	6.82444	0.5396
49	1VSG (A)	Coiled coil	2.9	364	6.55718	0.511659
50	1YGE	α+β	1.4	838	7.39878	0.569634
51	2AAK	α+β	2.4	152	5.01924	0.539899
52	2ACE	α/β	2.5	537	6.50626	0.516087
53	2AZA	β	1.8	129	4.70821	0.541237
54	2BUK	β	2.4	196	5.276	0.530857
55	2BW4 (A)	β	0.9	340	8.65057	0.501682
56	2CCY (A)	α	1.6	128	4.73291	0.540939
57	2CPK (E)	α+β	2.7	350	6.65625	0.522431
58	2CYP (A)	α	1.7	294	6.82446	0.53788
59	2GMF (A)	α	2.4	127	4.71749	0.536367
60	2HAD	α/β	1.9	310	6.66754	0.529489
61	2LIV (A)	α/β	2.4	344	6.49311	0.515426
62	2RN2	α/β	1.4	155	5.1163	0.536626
63	2TMV(P)	α	2.9	158	5.1163	0.536626
64	3CD4 (A)	β	2.2	182	5.26001	0.530102
65	3CHY	α/β	1.6	128	4.73524	0.541913
66	3CLA	α/β	1.7	213	5.80401	0.538686
67	3DFR	α/β	1.7	162	5.06915	0.534101
68	3GRS (A)	α+β	1.5	478	6.54601	0.520539
69	3PGK (A)	α/β	2.5	416	6.59167	0.520505
70	3PMG (A)	α+β	2.4	561	6.75883	0.517049
71	4BLM (A)	α+β	2.0	265	6.91026	0.543002
72	4GCR (A)	β	1.5	174	5.26087	0.529704
73	5FBP (A)	α/β	2.1	335	6.65569	0.529469

74	5P21	α/β	1.3	166	6.50332	0.536046
75	5PEP (A)	β	2.3	326	10.6363	0.625245
76	6ABP (A)	α/β	1.6	306	6.75971	0.532318
77	8ACN (A)	α/β	2.0	754	7.5381	0.512647
78	8ADH (A)	β	2.4	374	6.63043	0.511267
79	8ATC (B)	α+β	2.5	153	6.6688	0.529743
80	8ATC (A)	α/β	2.5	310	4.88002	0.541068
81	5LDH	α/β	2.7	334	6.42807	0.509017
82	4APE	β	2.1	330	6.37838	0.509181
83	8TLN	α+β	1.6	316	6.12938	0.509817
84	9API	α+β	3.0	347	6.42889	0.507465
85	2HIP	Small proteins	2.5	72	3.19437	0.503867
86	1BPL	β	2.2	189	4.54341	0.528325
87	1HDD	α	2.8	61	3.58333	0.557100
88	1I1B	β	2.0	153	4.87005	0.523601
89	1PGX	α+β	1.6	83	3.16356	0.505562
90	1SBT	α/β	2.5	275	5.19883	0.507135
91	1TAB(I)	β	2.3	82	3.33033	0.562105
92	1TIM	α/β	2.5	247	4.58732	0.508150
93	2C12	α	2.1	439	5.84653	0.499146
94	2CAB	β	2.0	260	4.88093	0.508475
95	2HMQ	α	1.6	114	3.89416	0.534960
96	2PAB	β	1.8	127	3.98657	0.534725
97	2SOD	β	2.0	152	4.78279	0.523628
98	351C	α	1.6	82	3.6963	0.539269
99	3TMS	α+β	2.1	264	4.94842	0.507948
100	5CNA	β	2.0	237	4.60206	0.509583

Chapter 3

Results and Discussion

In this chapter we present our analysis of the modularity based graph spectral approach (given in chapter 2) for identifying the number of domains and the domain boundaries in protein structures. The analysis has been carried out on the dataset of 100 protein structures (given in Table 2.2), consisting of both single and multi domain proteins. The multi-domain proteins analyzed include contiguous (Figure 3.1(a)) as well as non-contiguous domains along the polypeptide chain (Figure 3.1(b) and (c)).

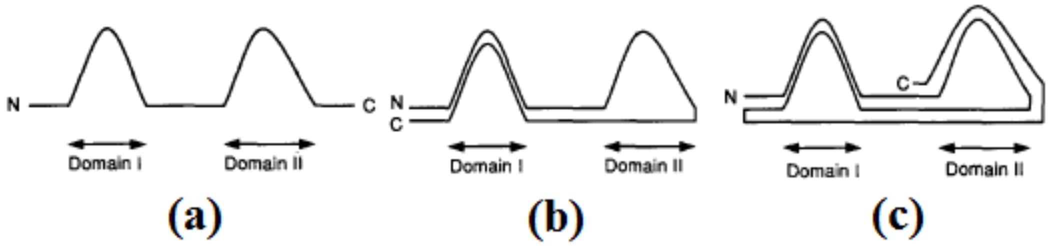


Figure 3.1: (a) Contiguous domains (b) One non-contiguous domain (c) Two non-contiguous domains (Siddiqui and Barton, 1995)

This dataset also includes a set of 55 proteins compiled by Jones *et al* for domain prediction accuracy (Jones *et al,* 1998), comprising of 34 single domain proteins, 17 two domain, 1 three domain and 3 four domain proteins. Since prediction accuracies of a number of domain identification methods such as DOMAK (Siddiqui and Barton, 1995), DomainParser (Xu *et al,* 2000) and DDomain (Hongyi Zhou *et al*, 2007) are available on this dataset, we also present the accuracy of domain prediction by our approach on this Jones *et al* dataset. The domain prediction on the complete set of 100 proteins is compared with the domain annotations provided in the CATH database (Orengo *et al,* 1997), and a web-based program, DomainParser (Xu *et al*, 2000). The domain assignment in CATH database is based on the topological structure instead of functional aspects as it employs a purely structural definition for domains (based on compactness) making it an appropriate database for comparing our modularity-based approach, which also find groups/clusters based on structural topology. Also the annotations in the CATH database being manually curated are more reliable. DomainParser algorithm, similar to out approach involves analysis of graph constructed by considering two residues in contact if the distance between their closest atoms is ≤ 4 Å. The protein residual network is divided into two parts by finding the minimum cut, that is, the cut that minimizes the total number of cross-edges between two groups using classical Ford Fulkerson algorithm (Ford and Fulkerson, 1956). The multi-domain decomposition problem is solved by repeatedly solving a series of two domain problems. The updated version of the program further refines the domain assignment by assessing the protein structure based on neural network approach that includes various parameters such as (i) the fitness of the hydrophobic moment profile of the predicted domain with the general hydrophobic moment profile of experimental structures, (ii) the number of nonconsecutive sequence segments in a predicted domain, (iii) the compactness of the domain, (iv) the size of the interface between the two domains, (v) relative motion between two domains, and (vi) the size of the predicted domain. The major limitation with the graph based partitioning approach in DomainParser is that it would always divide the network into two groups regardless of whether a good division exists or not. As a consequence it fails to identify single domain proteins. That is, the DomainParser program predicts the number of domains and its boundaries only for multi-domain proteins. Another recent graph based approach by Sistla *et al* (2005) identifies domains by the spectral analysis of the Laplacian matrix. Though their method works reasonably well for contiguous multi-domain proteins, we show that it fails to identify non-contiguous domains and also single domains. With about 25% of multi-domain proteins being non-contiguous and a reasonably large number of single domain proteins, there exists a need for reliable domain identification methods.

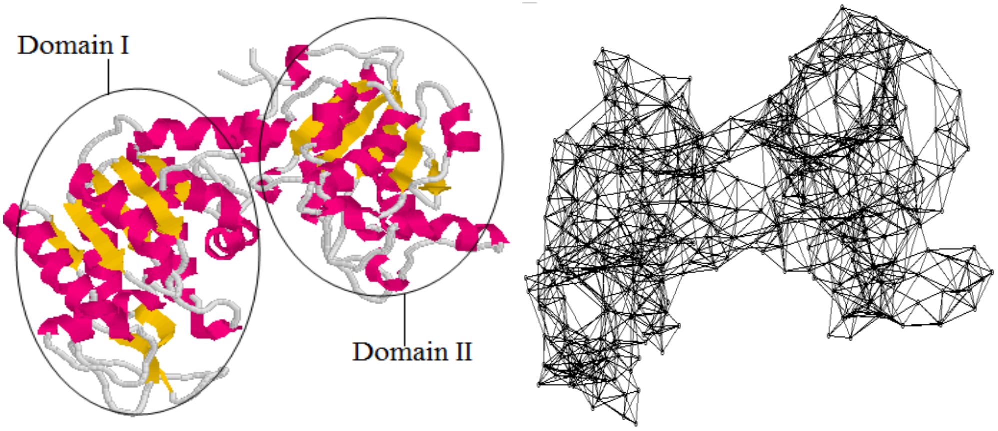
3.1 Identification of Multiple Domains

Below we first present the implementation of our approach in detail on three multi-domain proteins, *viz*., a contiguous 2-domain protein, 16PK, a non-contiguous 2-domain protein, 1HLE, and a 4-domain protein 8ACN. The analysis has been carried out on a large number of multi-domain proteins and the results are compared with annotation in CATH database, results obtained on the web-server, DomainParser and the graph spectral approach by Sistla *et al*. These results are summarized in the Table-17.

3.1.1 Identification of Contiguous Domains

First we present in detail the implementation of our approach on a 2-domain protein, 16PK(A), a phosphoglycerate kinase from *Trypanosoma brucei bisubstrate* analog. It is known to play a major role in the Calvin Cycle (carbon fixation stages). The 3-dimensional structure of 16PK (A) and its protein contact network are shown in Figure 3.2 and it belongs to α/β class. Figure 3.2(a) shows two similar alpha-beta sandwich domains, with domain I starting from the N-terminus. A plot of the distance matrix, constructed by computing the distance between the C_α atoms of all (i , j) pairs of residues (shown in Figure 3.3), clearly shows two distinct and compact structural modules as dark blue patches from 1 to 190 and 210 to 400. The protein contact network in Figure 3.2(b) is constructed by drawing an edge between pairs of C_α atoms that are within a 7Å distance and the network plot also clearly shows two distinct compact structural modules in the 3-dimentional space.

To automate the identification of domains, we have implemented two approaches: Newman's modularity based community detection algorithm - a top-down approach for identifying compact structural modules, followed by Clauset's greedy optimization algorithm - a bottom-up approach for agglomeration of the structural modules of compact secondary structural elements to identify true domains.



(a) (b)

Figure 3.2 (a) The 3-dimensional structure of 16PK(A), and (b) its network graph.

Table 3.1 summarizes the various steps in the decomposition of the protein contact network of the protein 16PK by the Newman's approach. At each subdivision of the graph, the modularity value is given along with the size and number of clusters formed at each split. The modularity value is seen to increase with every split and has a maximum value for the third split after which the modularity value starts falling again.

The clusters corresponding to the highest modularity value are C1 (**1-20, 183-197**), C2 (**198-404**), C3 (**61-84, 130-182**) and C4 (**21-60, 85-129**). It is observed that domain-II in Figure 3.2(a) is correctly identified as cluster C2 from 198-406, however the domain-I from 1-200 has been split into three clusters, comprising of non-contiguous peptides. That is, the Newman's modularity approach is seen to over cut a loosely packed domain into smaller closely packed secondary structure arrangements. To correctly identify the domains we now discuss the second phase of our approach which involves constructing a coarse protein structure graph of these structural modules. This is done by first considering the clusters corresponding to the split having maximum modularity value (in this case split 3 with four clusters: C1, C2, C3 and C4) as nodes and weighted edges drawn between them based on the number of interactions (edges) between the residues in each pair of clusters (eqn. 2.20). On this weighted graph Clauset's amalgamation approach is implemented.

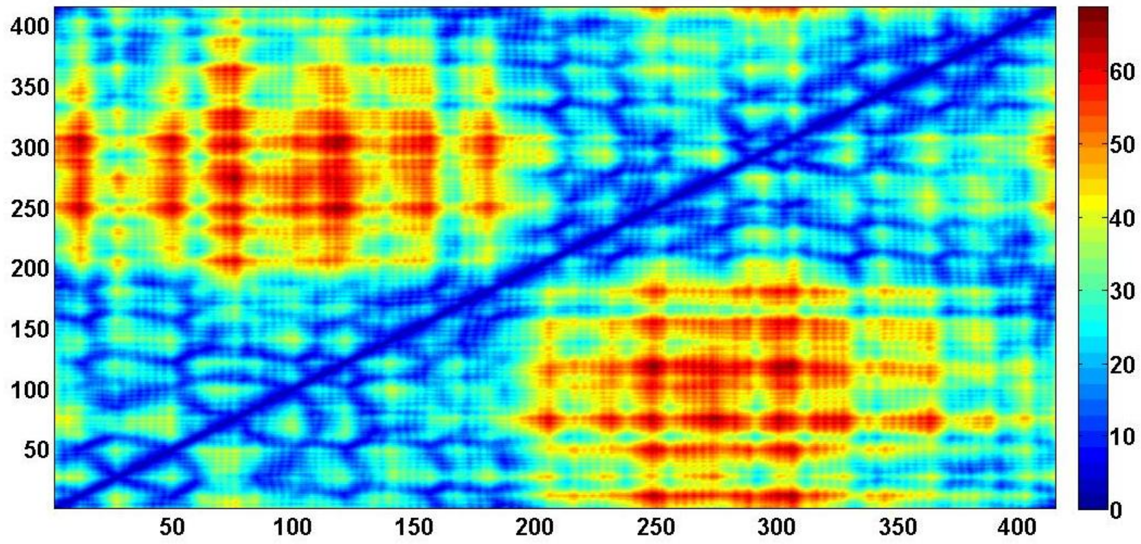


Figure 3.3: Contour plot of the distance matrix of the protein 16PK(A)

Table 3.1: Subdivision of protein contact network of 16PK (A)

Split No.	Modularity	No. of Clusters	Start/End of Clusters
1	0.485628	2	C1: 1-197 C2: 198-404
2	0.571309	3	C1:1-60, 85-129, 183-197 C2: 198-404 C3: 61-84, 130-182
3	**0.593277**	**4**	**C1: 1-20, 183-197** **C2: 198-404** **C3: 61-84, 130-182** **C4: 21-60, 85-129**
4	0.585292	5	C1: 1-20 C2: 198-404 C3: 61-84, 130-182 C4: 20-60, 85-129 C5: 183-197

In Table 3.2 the interaction matrix between the four clusters, C1, C2, C3, C4 corresponding to Split - 3 in Table 3.1 is given. The ij^{th} element corresponds to the number of interactions between clusters *i* and *j*, computed using equation 2.20. The value zero between C2 and C4 indicates no edges between them. As the pair C1 - C4 has the highest value, we expect them to be regrouped. However, cluster C2 has minimum interactions with the remaining three clusters C1, C3 and C4.

Table 3.2: Interaction matrix for clusters in Split-3 of Table 3.1 for 16PK(A)

	C1	**C2**	**C3**	**C4**
C1	**0**	**55.77**	**198.34**	**305.34**
C2		**0**	**38.73**	**0**
C3			**0**	**158.53**
C4				**0**

To find out which of these three clusters would re-group and in what order to form single or multiple domains, Clauset's modularity based greedy optimization algorithm is implemented on the coarse protein structure graph. The agglomeration steps in this approach are shown in Table 3.3. The structural modules given by the second merging step with the highest modularity value are predicted as true domains: D1 = {C1 (1–20, 183-197), C3 (61-84, 130-182), C4 (21-60, 85-129)}, i.e., D1 = {**1-197**} and D2 = {**198-404**}.

The graph spectral method by Sistla *et al* was also implemented on the protein 16PK. In Figure 3.4 the plot of sorted eigenvector component of second lowest eigenvalue, 2sev, of the Laplacian matrix (in blue) for protein contact network of 16PK is shown. The eigenvector plot clearly shows two distinct plateau regions suggesting two domains.

Table 3.3: Amalgamation of protein clusters from Split-3 in Table 3.1 for protein 16PK (A)

Merge No.	Modularity	No. of Clusters	Agglomeration of Clusters
1	- 0.302817	3	D1:C1, C4 D2: C2 D3: C3
2	**0.125917**	**2**	**D1: C1 C3, C4** **D2: C2**

The domain boundaries are identified by the intersection of the slope of this curve (shown in black), which intersects with the plot of 2sev, at 208. Thus, the two domains predicted by this approach are: D1 = {1 - 208} and D2 = {209 - 404}. Our predictions of domain boundaries for the protein 16PK are summarized in Table 3.4. A very good agreement of our predictions is observed with annotation in CATH database, output from DomainParser and predictions by graph spectral approach of Sistla *et al.*

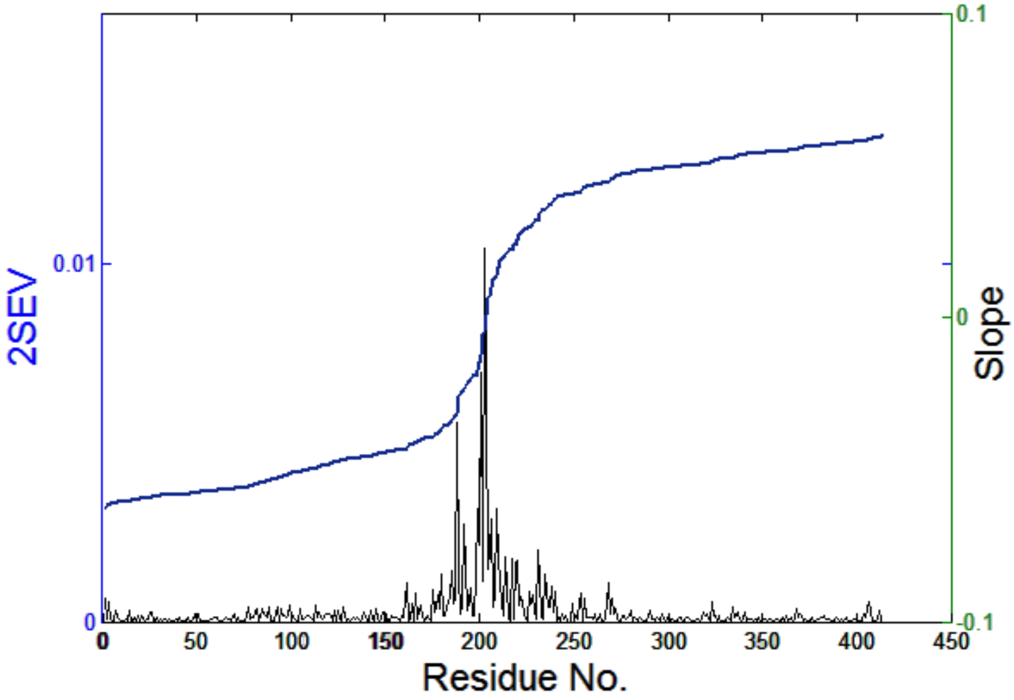


Figure 3.4: Eigen vector corresponding to the second lowest eigenvalue and its slope for 16PK(A)

Table 3.4: Comparison of domain predictions by various approaches for protein 16PK(A)

PDB ID	Class	CATH Annotation	DomainParser Output	Sistla *et al*'s approach	Our approach
16PK (A)	α/β	I: 5-192 II: 199-406	I: 5–204 II: 205–406	I: 1 to 208 II: 209 to 406	I: 1-197 II: 198-404

The proposed approach also works for proteins containing more than two domains and we next present our analysis on a 4-domain protein, 8ACN(A), a aconitase enzyme from *Bos taurus*. It converts citrate to isocitrate in Aconitate Hydratase pathway. Its 3-dimensional structure is given in Figure 3.5 (a) and the corresponding protein contact network in Figure 3.5(b) and it also belongs to and belongs to mainly α/β class. From a visual analysis of the C_α-C_α distance matrix plot of the protein 8ACN(A) in Figure 3.6, we observe four dark blue patches from 1-200, 201-300, 301-500, and 550-750 along the central diagonal suggesting four compact structural modules in this protein structure. Thus a quantitative idea of the number of domains and their approximate boundaries can be obtained from the visualization of distance matrix plot when the domains are contiguous. Table 3.5 summarizes the modularity value for each successive subdivisions of the protein contact network of 8ACN by Newman's approach. The fourth split has the highest modularity value and hence the clusters C1 - C5 are used for constructing the weighted protein structure graph. The regrouping of these five clusters by Clauset's amalgamation approach is given in Table 3.6 and the first merging step corresponds to the highest modularity value, thus predicting four structural domains in chain A of protein 8ACN: D1 = {C1 (1-204)}, D2 = {C2 (205-321)}, D3 = {C3 (322-497)} and D4 = {C4,C5 (498-754)}. This prediction is in agreement with the CATH annotation, shown in Table 3.7. No results were obtained from DomainParser for this protein.

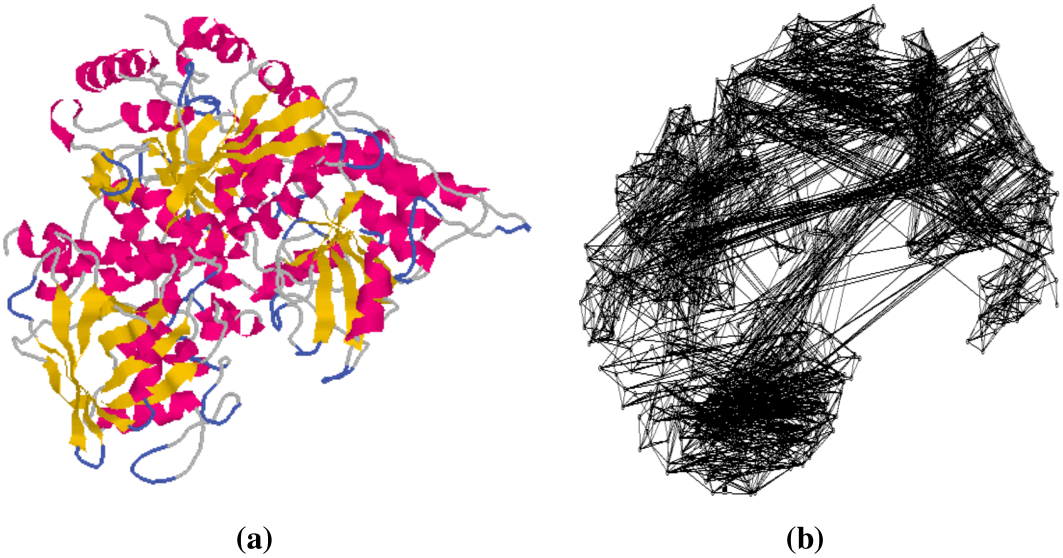


Figure 3.5: (a) The 3-dimensional structure of 8ACN(A), and (b) its network graph.

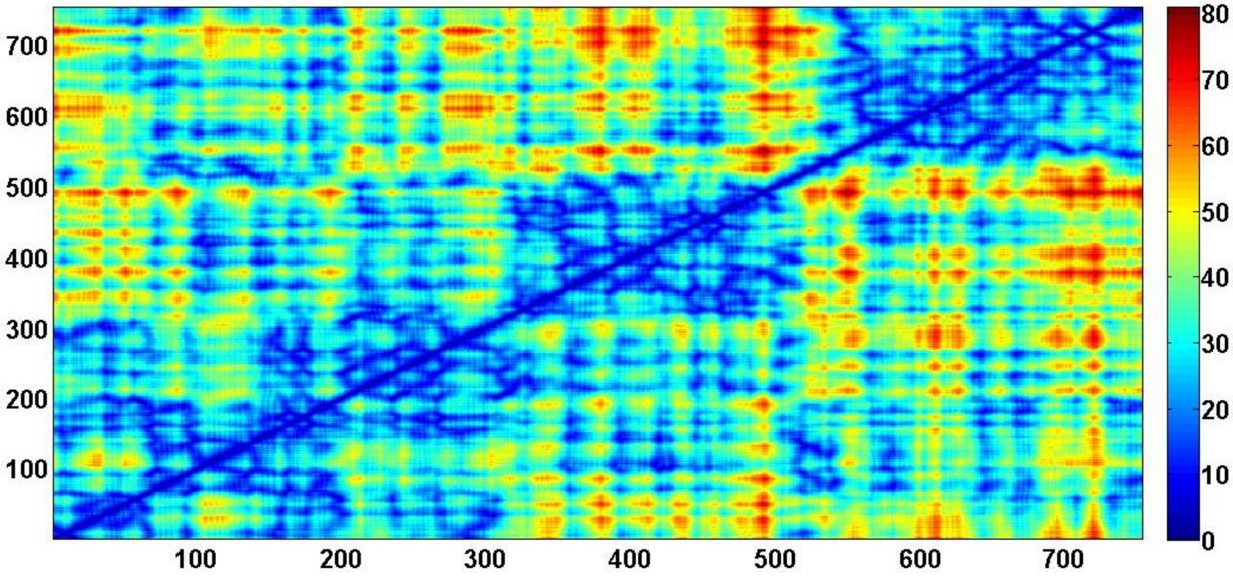


Figure 3.6: Contour plot of the distance matrix for protein 8ACN(A)

Table 3.5: Subdivision of protein contact network of 8ACN (A)

Split No.	Modularity	No. of Clusters	Start/End of Clusters
1	0.366672	2	C1: 1-497 C2: 498-754

2	0.623382	3	C1: 1-204, 322-497 C2: 205-321 C3: 498-754
3	0.65763	4	C1: 1-204 C2: 205-321 C3: 322-497 C4: 498-754
4	**0.658602**	**5**	**C1: 1-204** **C2: 205-321** **C3: 322-497** **C4: 498-560** **C5: 561-754**
5	0.651494	6	C1: 1-73 C2: 205-321 C3: 322-497 C4: 498-560 C5: 561-754 C6: 74-204

Table 3.6: Amalgamation of Clusters in Spit-4 of Table 3.5 for 8ACN(A)

Merge No.	Modularity	No. of Clusters	Agglomeration of Clusters
1	**0.372604**	**4**	**D1: C1** **D2: C2** **D3: C3** **D4: C4, C5**
2	0.031926	3	D1: C1, C3 D2: C2 D3: C4,C5
3	-0.119732	2	D1: C1, C2, C3 D2: C4,C5

Table 3.7: Comparison of CATH annotation, DomainParser results with our spectral method for the protein 8ACN(A)

PDB ID	Class	CATH Annotation	DomainParser Output	Our Approach
8ACN (A)	α/β	I: 2-202 II: 203-315 III: 316-490 IV: 534-754	Error Message: Cannot Parse 8ACN(A)	I: 1-204 II: 205-321 III: 322-497 IV: 498-754

For multi-domain proteins, when the domains are very closely packed in the 3-dimensonal space, we observe that our approach fails to correctly identify the domain boundaries. In such situations, manual inspection is necessary. It may be noted that manual curation is an important and reliable feature of domain assignment in CATH database. For instance, when our approach is implemented on a 4 domain protein, 1CDG(A), transferase enzyme from *Bacillus circulans* the domain boundaries predicted are: D1 = {C2 (1-142, 155-165, 213-234, 250-395)}, D2 = {C3 (**396-502**)}, D3 = {(C1, C5) = **503 – 584**} and D4 = {C4 (143-154, 166-212, 235-249, 585-688)}. As shown in the Figure 3.7, the domain D1 has short regions intertwined (in 3-D space) with domain D4 from 1 – 395. Parts of the short non-contiguous regions in Domain D4, (encircled in Figure 3.7(b)) are actually part of the domain D1. So by visual inspection we assign these three short regions as part of D1 and after curation the domain boundaries are: D 1= 1-393, D2 = 396-502, D3 = 503-584 and D4 = 585-688. Thus, it is very important to manually check the domain assignment by any computational approach.

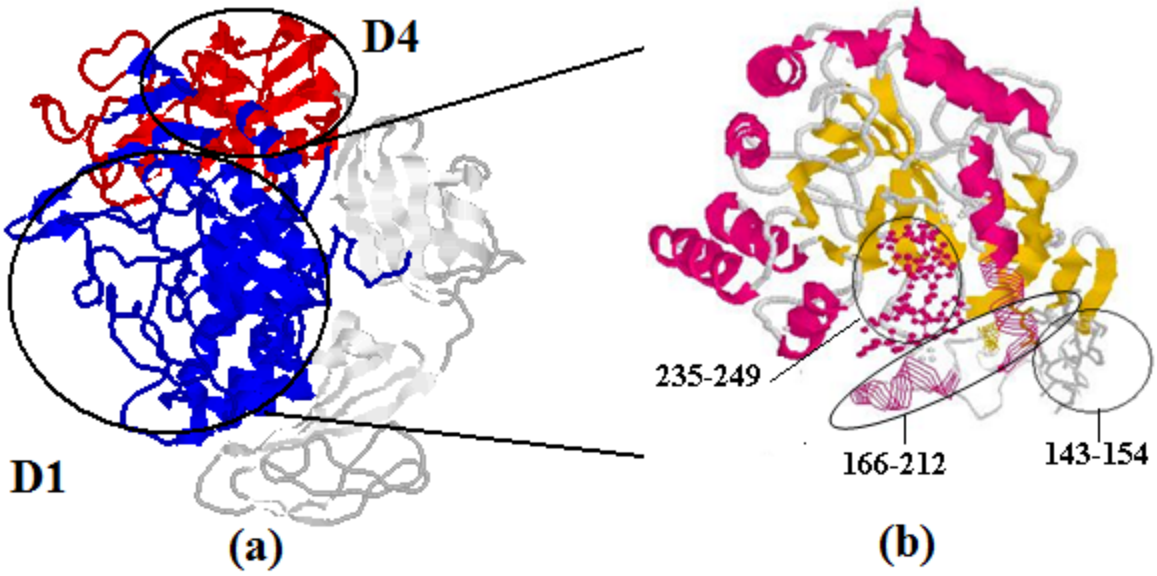


Figure 3.7: (a) 3-dimensional structure of the protein 1CDG, (d) 3-dimentional structure of the region 1-395.

Table 3.8: Comparison of CATH annotation, DomainParser results with our spectral method for the protein 1CDG(A)

PDB ID	Class	CATH Annotation	DomainParser Output	Our Approach
1CDG (A)	Mainly β	I: 1-400 II: 401-495 III: 496-582 IV: 582-685	Error Message: Cannot Parse 1cdg A	I: 1-395* II: 396-502 III: 503-584 IV: 585-688*

3.1.2 Identification of Non-contiguous Domains

Protein 1HLE is a hydrolase inhibitor (serine proteinase) from *Equus caballus* and contains two domains which are non-contiguous along the polypeptide chain. It also belongs to α/β class and its 3-dimentional structure is shown in Figure 3.8(a). The N-terminus domain I, enclosed by the red rings is made up of two discontinuous regions (1-26 and 171-276) comprising of 7 α-helices (pink color), while Domain II, enclosed by the blue rings, is the larger domain extending to the C-terminus, and is formed by the discontinuous regions (27-170 and 276-341) consisting mainly β-sheets (yellow color) and one α-helix (pink color). The C_{α}-C_{α} distance matrix plot of the protein 1HLE(A) is shown in Figure 3.9. Three compact modules can be identified as blue patches from 30 – 90, 100 – 160, 170 – 240 along the central diagonal, but protein contact network as shown in Figure 3.8(b) has two spatially compact regions corresponding to the number of domains which is in agreement with CATH annotations. Thus in such cases where we have non-contiguous domains, by using distance matrix plots it is difficult to identify correctly the true domains.

By analyzing the modularity value of every split in the partitioning of the network, it is observed that the 4th split with 5 clusters has the highest modularity value in Table 3.9 and is considered for the agglomeration phase. On implementing Clauset's agglomeration approach on these 5 clusters, the third merging step has the highest modularity value, predicting two domains as: D1 = {C1 = **1–26**, C4, C5 = **171–275**} and D2 = {C2 = **27–170**, C3 = **276–341**} (in Table 3.10).

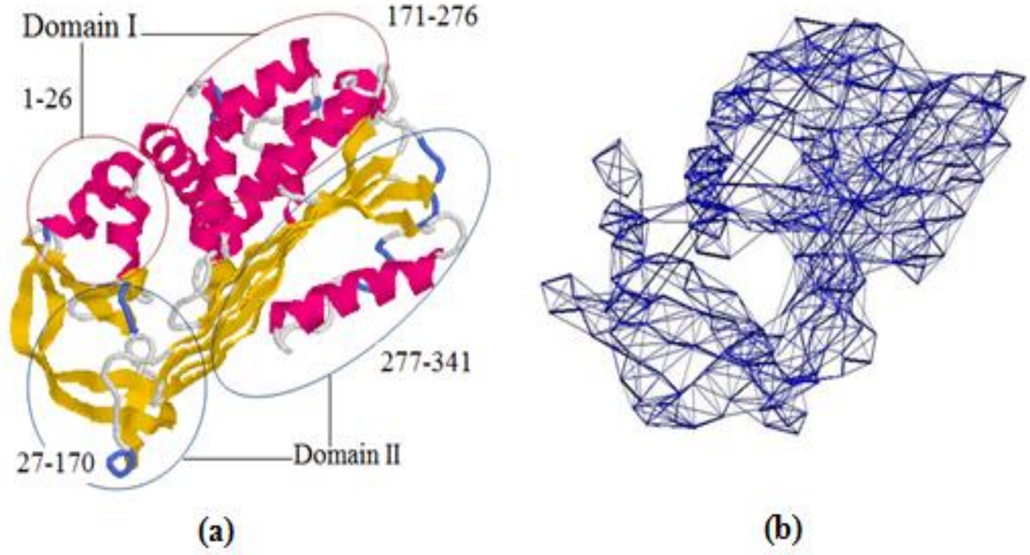


Figure 3.8: (a) The 3-dimensional structure of 1HLE(A), and (b) its network graph.

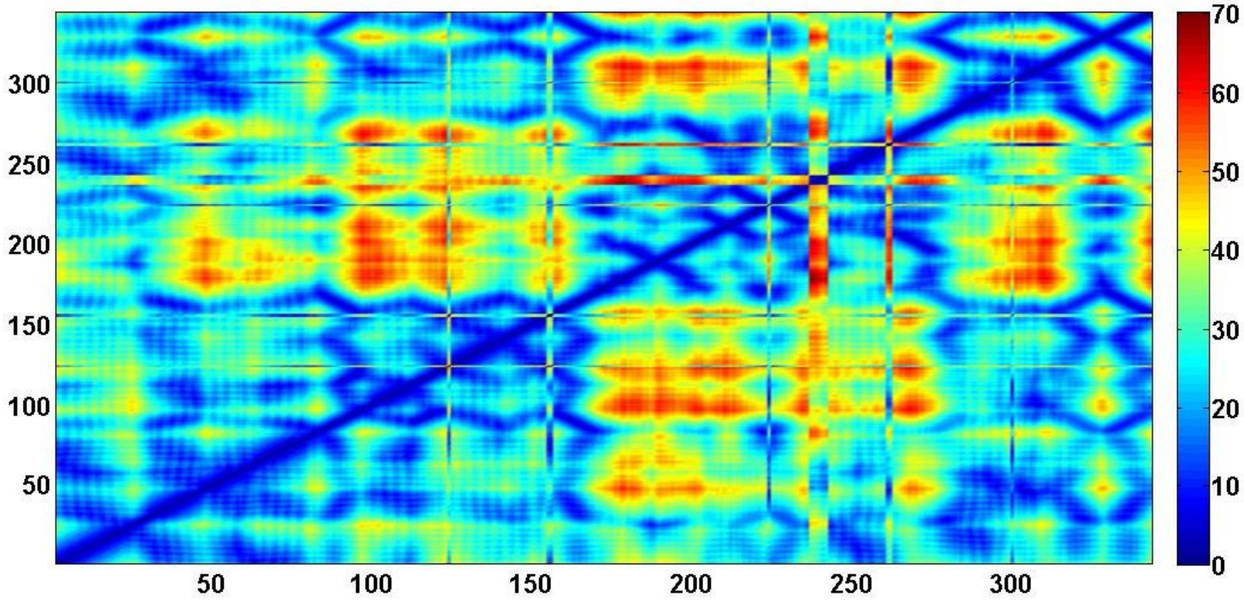


Figure 3.9: Contour plot of the distance matrix of the protein 1HLE(A)

Table 3.9: Subdivision of protein contact network of 1HLE (A)

Split No.	Modularity	No. of Clusters	Start/End of Clusters
1	0.412365	2	C1: 1-26,42-76, 98-125, 171-275, 287-314 C2: 27-41, 77-97, 125-170, 276-286, 315-341
2	0.583516	3	C1: 1-26, 171-275 C2: 27-41, 77-97, 116-170, 276-286, 315-341 C3: 42-76, 98-115, 287-314
3	0.634473	4	C1: 1-26, 243-260 C2: 27-41, 77-97, 116-170, 276-286, 315-341 C3: 42-76, 98-115, 287-314 C4: 171-242, 261-275
4	**0.638779**	**5**	**C1: 1-26** **C2: 27-41, 77-97, 116-170, 276-286, 315-341** **C3: 42-76, 98-115, 287-314** **C4: 171-242, 261-275** **C5: 243-260**
5	0.634579	6	C1: 1-12 C2: 27-41, 77-97, 116-170, 276-286, 315-341

			C3: 42-76, 98-115, 287-314 C4: 171-242, 261-275 C5: 243-260 C6: 13-26

Table 3.10: Amalgamation of clusters in Split-4 of Table 3.9 for 1HLE (A)

Merge No.	Modularity	No. of Clusters	Agglomeration of Clusters
1	- 0.227604	4	D1: C2, C3 D2: C1 D3: C4 D4: C5
2	- 0.0221965	3	D1: C1, C5 D2: C2, C3 D3: C4
3	**0.0564953**	**2**	**D1: C1, C4, C5** **D2: C2, C3**

It may be noted that both the domains are made up of non-contiguous regions along the polypeptide chain. When the graph spectral approach by Sistla *et al* was implemented on this protein, it failed to predict the domains correctly. Figure 3.10 shows the plot of the sorted eigenvector components of the second lowest eigenvalue, 2SEV (shown in blue) of Laplacian matrix and its slope (shown in red) for the protein 1HLE. It is observed that there is only one plateau region (from 1 – 150) in the slope plot and it intersects the 2SEV curve at a number of places making it impossible to make any predictions for domain boundaries. Our predictions of domain boundaries for 1HLE(A) are compared with the annotation in CATH database, DomainParser output and the prediction by Sistla *et al* in Table 3.11. A good agreement with the annotation in CATH database and with the output of DomainParser is observed. The Sistla *et al*'s approach fails to make any predictions in this case when the domains are non-contiguous.

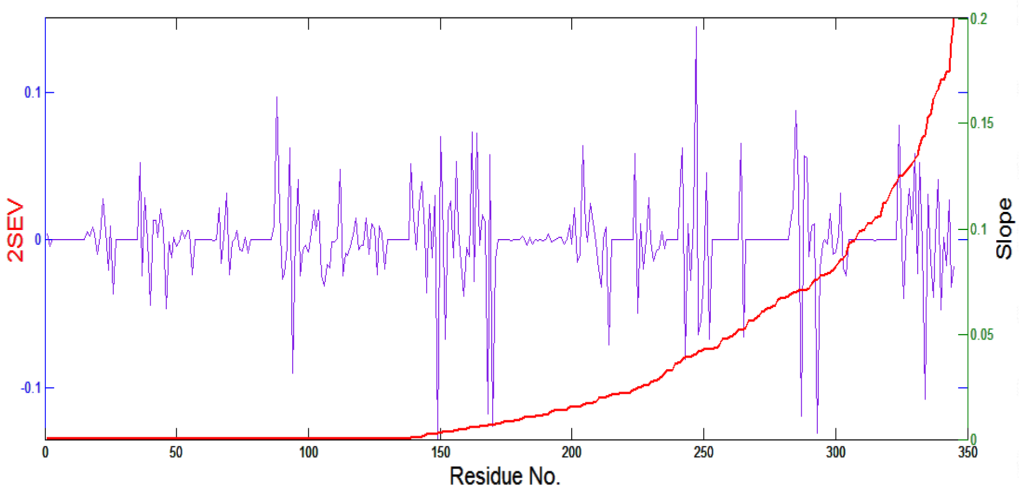


Figure 3.10: Eigen vector corresponding to the second lowest eigenvalue and its slope for 1HLE(A)

Table 3.11. Comparison of domain predictions for protein 1HLE (A).

PDB ID	Class	CATH Annotation	DomainParser Output	Sistla *et al*'s Approach	Our approach
1HLE (A)	α/β	I: 23-193, 290-358 II: 194-289	I: 23-203, 290-358 II: 204-289	No Prediction	I: 27-170, 276-341 II: 1-26, 171-289

Table 3.12 summarizes our analysis on a set of 10 proteins from different structural classes (α, β, α+β, α/β) having non-contiguous domains. The results of our approach are compared with that of the graph spectral approach of Sistla *et al*, DomainParser and annotations in CATH. It is observed that for most proteins with non-contiguous domains, the DomainParser is unable to identify domains and gives an error message. Also, the graph spectral approach by Sistla *et al* fails to correctly identify the domains in each case, while there is a very good agreement not only in the number of domains predicted but also in the prediction of domain boundaries by our graph spectral approach, except for the protein, 1PHA(A). However, Siddiqui and Barton (1995) using their approach, DOMAK, and by visual inspection have predicted two non-contiguous domains as shown in Figure 3.11. The domain boundaries predicted by them are: D1: 10-110 (in red), 296-355 (in

green) and D2: 102-295 (in blue), 355-414 (in orange). Our predictions are in accordance with that of Siddiqui and Barton for protein 1PHA(A).

Table 3.12: Comparison of domain predictions in non-contiguous proteins.

S. No.	PDB ID	Class	CATH Annotation	*Sistla et al*'s Approach	DomainParser Output	Our approach
1	1LFI (A)	α/β	I: 1-91, 251-339 II: 92-250 III: 340-434, 595-691 IV: 435-594	No prediction	Error Message	I: 1-86, 257-339 II: 87-256 III: 340-441, 605-691 IV: 441-605
2	1PHA (A)*	α	I: 10-414	I:25-414	Error Message	I: 1- 108, 303-364 II: 109-302, 365-414
3	2CPK (E)	α+β	I: 15-32, 127-317 II: 33-126, 318-350	I: 20-350	Error Message	I: 1-42, 132-321 II: 43-131, 322-350
4	2LIV (A)	α/β	I: 1-120, 250-328 II: 121-249, 329-344	No prediction	Error Message	I: 1-118, 246-329 II: 119-245, 330-344
5	6ABP (A)	α/β	I: 2-108, 255,285 II: 109- 254, 286-306	I: 20-306	Error Message	I: 1-110, 254-291 II: 111-253, 292-306
6	1PRC (C)	A	I: 24-142, 314-332 II: 143-205, 212-313	I: 75-310	Error Message	I: 1-45, 146-310 II: 46-145, 311-336
7	1SGT (A)	B	I: 16-31, 125-236 II: 32-124, 237-245	I: 38-245	Error Message	I: 16-29, 133-236 II: 30-132, 237-245
8	1HLE (A)	α/β	I: 23-193, 290-358 II: 194-289	No prediction	I:23-203, 290-358 II: 204-289	I: 27-170, 276-341 II: 1-26, 171-289
9	2CPY (A)	A	I: 4-144, 266-294 I: 145-265	I:1-265	Error Message	I: 1-144, 260-294 II: 145-259
10	1GPB	α/β	I: 19-485, 813-836 II: 486-812	I: 40-330 II: 331-810	Error Message	I: 19-492, 811-836 II: 493-810

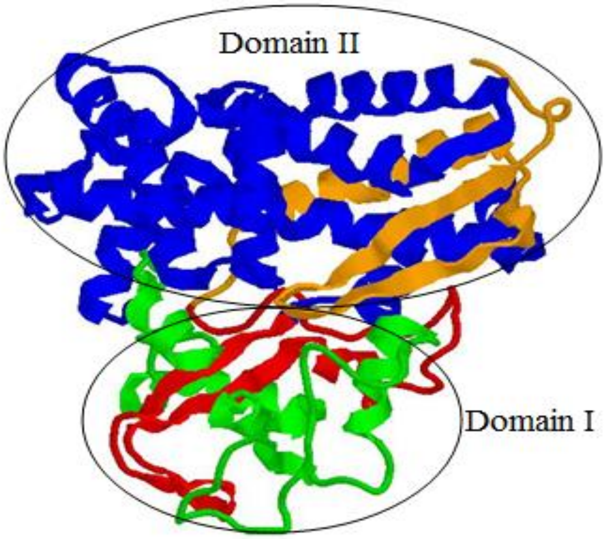


Figure 3.11: 3-Dimentional structure of 1PHA(A)

Table 3.14 summarizes the analysis of 40 multi-domain proteins (excluding the 10 non-contiguous multi-domain proteins analyzed in Table 3.12) using the graph-based modularity approach. The first 21 proteins tabulated are taken from Jones *et al* dataset. The predictions are ompared with the annotations in CATH database and the output from DomainParser. It is observed that for a large number of proteins, DomainParser failed to give any output while a good agreement of our domain predictions is observed with the annotations in CATH database. Out of the 40 proteins in this dataset, correct predictions have been made for 39 of them. The one exception, 1YGE is reported as a 5 domain protein in CATH database. But our approach predicted only three domains – D1, D2 and D3. The domain D2 predicted by us is split into three domains in the assignment in CATH database. In the top-down phase of our approach, the domain D2 is divided into 3 clusters (C2, C3 & C4). But as the number of interactions across these closely packed clusters are quite strong (see interaction matrix given in Table 3.13), the agglomeration step merges them into a single domain as shown in Figure 3.12.

Table 3.13: Interaction matrix for protein 1YGE.

	C1	C2	C3	C4	C5
C1	0	95.34	0	76.42	0
C2		0	210.54	253.82	0
C3			0	218.25	98.34
C4				0	87.94
C5					0

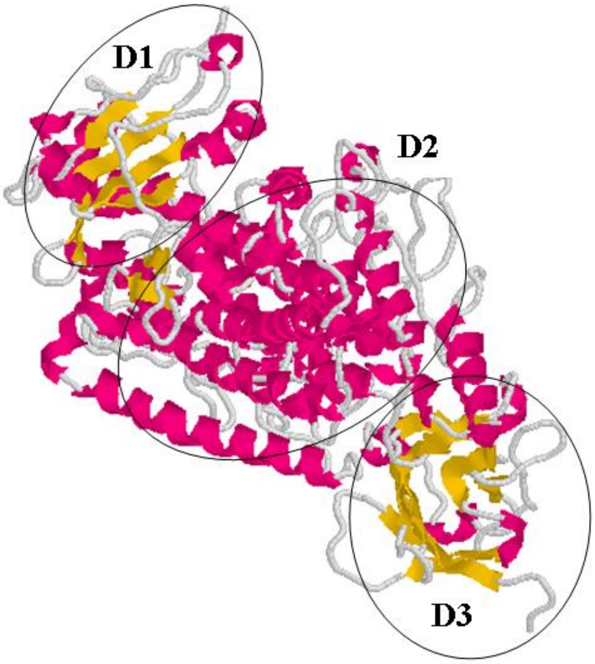


Figure 3.12: 3-dimentional structure of 1YGE.

Table 3.14: Comparison of domain predictions for multi-domain proteins.

S.No.	PDB ID	Class	CATH Annotation	DomainParser Output	Our Approach
1	1EZM (A)	α+β	I:1-152 II:153-298	Cannot parse PDB structure	I: 1-145 II: 146-298
2	1FNB (A)	β	I:19-151 II:152-314	Cannot parse PDB structure	I: 1-148 II: 148-314
3	1RHD (A)	α/β	I:1-156 II:157-293	Cannot parse PDB structure	I: 1-154 II: 155-293
4	1G6N (A)	α	I:9-137 II:138-206	Cannot parse PDB structure	I:1-136 II:137-206
5	1GPD (A)	α/β	I:1-147, 314-333 II:148-313	Cannot parse PDB structure	I: 1-144, 313-333 II: 145-312
6	1LAP (A)	α/β	I:1-165 II:166-483	Cannot parse PDB structure	I: 1-165 II: 166-483
7	3PGK (A)	α/β	I:1-186 II:187-401	I: 0-188;402-415 II: 189-401	I: 1-173 II: 174-401
8	4GCR (A)	β	I:1-83 II:84-174	Cannot parse PDB structure	I: 1-81 II: 82-174
9	3CD4 (A)	β	I: 1-98 II: 99-173	Cannot parse PDB structure	I: 1-103 II: 104-173
10	5FBP (A)	α/β	I:7-199 II:200-334	Cannot parse PDB structure	I: 1-201 II: 202-334
11	8ATC (B)	α+β	I:8-100 II:101-153	I: 8-97 II: 98-153	I: 1-100 II: 101-153
12	8ATC (A)	α/β	I:1-133, 292-310 II:134-291,	Cannot parse PDB structure	I: 1-136, 294-310 II: 137-293
13	8ADH (A)	β	I:1-178, 318-374 II:179-317	I: 1-173;321-374 II: 174-320	I: 1-174, 316-374 II: 175-315
14	1PFK (A)	α/β	I:1-141, 256-302 II:142-251, 303-	Cannot parse PDB structure	I: 1-144, 258-303 II: 145-257, 304-

			318		318
15	1VSG (A)	Coiled coil	I:1-33, 86-255 II:34-85, 256-362	I: 1-32;86-255 II: 33-85;256-362	I: 1-36, 89-255 II: 37-88, 256-362
16	1PHH (A)	α/β	I:1-72, 96-180, 269-351 II:73-95, 181-268, 352-388	Cannot parse PDB structure	I:1-71, 99-183, 269-351 II:72-98, 184-268, 352-388
17	3GRS (A)	α+β	I:18-160, 290-365 II:161-289 III:366-478	Cannot parse PDB structure	I:1-156, 287-368 II:157-286 III:369-478
18	8ACN (A)	α/β	I: 2-202 II: 203-315 III: 316-490 IV: 534-754	Cannot parse PDB structure	I: 1-204 II: 205-321 III: 322-497 IV: 498-754
19	1ATN (A)	α+β	I: 1-34, 71-134, 337-372 II: 35-68 III: 136-181, 271-332 IV: 182-267	I: 0-147; 337-372 II: 148-336	I: 1-38, 72-137, 335-372 II: 39-71 III: 138-185, 265-334 IV: 186-264
20	3PMG (A)	α+β	I: 1-197 II: 198-300 III: 301-400 IV: 401-561	I: 1-190 II: 407-561 III: 191-303 IV: 304-406	I: 1-201 II: 201-306 III: 307-397 IV: 398-561
21	1PPR (M)	α	I: 1-155 II: 156-312	I: 1-156, 198- 216, 262-277 II: 157-178, 217-261, 299-312	I: 1-25, 75 -155 II: 26-74, 156-312
22	1PGR (B)	β	I: 1-108 II: 109-213	I: 1-108 II: 109-213	I: 1-108 II: 109-209
23	1CDG (A)	Mainly β	I: 1-400 II: 401-495 III: 496-582 III: 582-685	Error Message: Cannot Parse 1cdg A	I: 1-395 * II: 396-502 III: 503-584 III: 585-688
24	2BW4 (A)	β	I: 7-158 II: 159-328	I: 2-310	I: 47-166 II: 167-375

25	1KF6 (B)	α/β	I: 2-105 II: 106-243	I: 1-107 II: 108-243	I: 1-89 II: 90-243
26	16PK (A)	α/β	I: 5-192 II: 199-406	Error Message: Cannot parse 16pk A	I: 1-197 II: 198-404
27	1D5M (B)	α+β	I: 4-81 II: 82-181	I: 2-92 II: 93-190	I: 1-86 II: 87- 181
28	1LAM (A)	α/β	I: 1-161 II: 162-484	I: 1-162 II: 163-484	I: 1-161 II: 162-484
29	1SMP (A)	β	I: 1-17, 250-471 II: 18-249	I: 4-15, 250-374 II: 16-56, 183-249 III: 375-471 III: 57-182	I: 1-12, 250-468 II: 13-249
30	5PEP (A)	β	I: 1-170 II: 171-327	No. of Ds: 1 Cannot Parse 5pep A	I: 1-185 II: 186-306
31	1BKS (B)	α/β	I: 9-53, 87-205 II: 54-86, 206-391	Error Message: Cannot parse 1BKS	I: 1-56, 93-210 II: 57- 92, 211-391
32	1YGE#	α+β	I: 9 -167 II:168 -267 III: 268 -356 IV: 357 -490 V: 491 -839	Cannot parse PDB structure 1yge chain A	I: 1-272 II: 273-494 III: 495-839
33	1SU4	α+β	I: 1-43,124-242 II: 44-123, 243-345, 746-994 III: 359-602 IV: 346-358, 603-745	I:1 - 16 ,147 - 241 II: 358 – 602 III: 323 – 357, 603 - 751 IV: 17 – 62, 101 – 146, 242 - 262 V: 63 – 100, 263 – 322, 752 - 994	I: 1-37, 125-146 II: 38-124, 247-339, 746-994 III: 340-605 IV: 606-745
34	1SMR (A)	β	I: 1-170 II: 171-325	I: 2-171 II: 172-326	I: 1-174 II: 175-335
35	1TAH	α/β	I:2-319	I: 2-319	I: 2-165 II: 166-319

36	1EPW	β	I: 1-438 II: 444-475, 532-830 III: 846-1078 IV: 1092-1282	I: 1 - 533 II: 534 - 861 III: 862 – 1077 IV: 1078 - 1290	I: 1- 443 II: 444-480, 534-831 III: 832-1075 IV: 1076-1290
37	5LDH	α/β	I: 1-164 II: 165-331	Error Message: Cannot parse 5LDH	I: 1-171 II: 172-334
38	4APE	β	I: 2-170 II: 171-326	Error Message: Cannot parse 4APE	I: 2-174 II: 175-329
39	8TLN	α+β	I: 6-154 II: 155-315	Error Message: Cannot parse 8TLN	I: 6-158 II: 159-316
40	9API	α+β	I: 195-289 II: 23-194, 290-358	Error Message: Cannot parse 9API	I: 192-291 II: 20-191, 292-358

3.2 Identification of Single Domains

As shown above, the proposed two-phase modularity-based graph spectral approach is able to correctly predict the number of domains as well as the domain boundaries in multi-domain proteins even if the domains are non-contiguous. However, it fails to distinguish between single and multi-domain proteins. First we explain with an example why the approach fails and then we propose modification of the approach by adding two filters for identifying single domain proteins. The split structure of single domain protein 2CCY by Newman's approach into closely packed groups of secondary structural elements is shown in Table 3.15. The second split with three clusters has the highest modularity value and is used as an input for the amalgamation phase, summarized in Table 3.16. There is only one merging step with C1 and C3 merging to give two domains, D1 {C1, C3 = **1-71**} and D2 {C2: **72-128**}. When we closely look at the 3D structure of 2CCY (Figure 3.13), we observe that residues 1-71 comprises of two closely packed helices which are slightly distant from the group of two helices formed by residues 72-128. Because of this spacing between the two groups of helices, the Clauset's algorithm fails to merge them into a single structural domain. Furthermore the limitation with the Clauset's algorithm is that the modularity measure always takes the maximum value '1' when all the clusters are grouped into a single cluster. Thus identifying single domains by this approach is not possible.

Table 3.15: Subdivision of protein contact network of 2CCY

Split No.	Modularity	No. of Clusters	Start/End of Clusters
1	0.325961	2	C1 : 1-58 C2 : 59-128
2	**0.345864**	**3**	**C1 : 1-58** **C2 : 72-128** **C3 59-71**
3	0.330463	4	C1 : 1-36, 43-58 C2 : 72-128 C3 59-71 C4: 37-42

Table 3.16: Amalgamation of clusters of Split-2 in Table 3.15 for protein 2CCY (A)

Merge No.	Modularity	No. of Clusters	Positions (D : Domain)
1	0.03963	2	D1 : C1, C3 D2 : C2

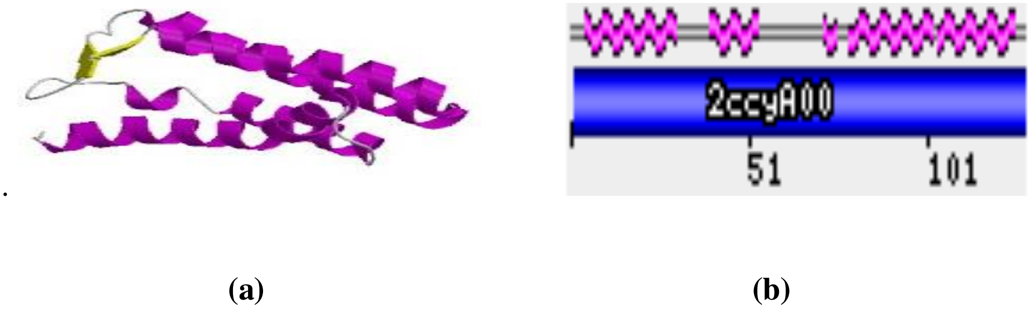


Figure 3.13: (a) 3-dimensional structure of 2CCY(A); (b) Secondary structure of 2CCY (A).

To overcome this problem we next investigated various graph-based properties such as diameter, density, clustering coefficient and interaction strength between two clusters obtained in the first split of Newman's modularity approach to see if any of these may help in differentiating between single and multi-domain proteins. These properties are computed on the data set of 100 proteins and the results are summarized in Table 3.16 and Table 3.17 for single domain and multi-domain proteins respectively. Since it has been observed that a protein larger than 250-300 amino acids usually consists of more than one structural domain (Cheng *et al*, 2006), we use length of the protein sequence (≤ 300) as the first filter to differentiate between single and multi domain proteins. Using this cut-off alone on the dataset of 100 proteins, only 18 proteins were wrongly classified: 5 single domain proteins had length > 300 (1GOX, 2ACE, 2HAD, 2C12, 1TAH) and 13 multi-domain proteins had length < 300 (8ATC(B), 4GCR, 3CD4, 1D5M, 1G6N, 1PGR, 1SGT, 1KF6, 1BKS(B), 1RHD, 1FNB, 2CYP, 1EZM). These proteins are marked with an asterisk in the Tables 3.17 and 3.18.

Table 3.17: Length, Density, Clustering coefficient and Split-1 interactions for Single Domain Proteins

PDB ID	Length	Density	Clustering Coefficient	Diameter	Split-1 Interactions
1BBH-A	131	0.1186	0.5339	11	**36**
1BBP-A	173	0.0803	0.5311	18	35
1BKS	255	0.0597	0.5165	17	43
1BRD	170	0.0824	0.5312	18	35
1FXI	96	0.1502	0.5671	10	17
1GKY	186	0.0755	0.5333	20	35
1GMP-A	96	0.1502	0.5671	10	17
1GOX*	350	0.0447	0.5136	20	29
1OFV	169	0.0831	0.5317	18	35
1PPN	212	0.0619	0.5171	14	37
1PYP	280	0.0659	0.5117	17	17
1RBP	174	0.0799	0.5302	18	35
1RCB	129	0.1207	0.5343	11	35
1RVE	244	0.0612	0.5227	19	33
1SNC	136	0.1129	0.5307	11	35
1TIE	166	0.0848	0.5298	18	35
1TLK	103	0.1444	0.5560	10	20
1ULA	289	0.0654	0.5109	16	19
2AAK	150	0.0978	0.5297	16	35
2ACE*	527	0.0307	0.5062	20	17
2AZA	129	0.1207	0.5343	11	35
2BUK	187	0.0753	0.5332	20	35
2CCY-A	127	0.1208	0.5374	11	25
2GMF-A	121	0.1262	0.5354	11	26
2HAD*	310	0.0521	0.5135	16	48

2RN2	155	0.0934	0.5303	19	35
2TMV-P	155	0.0934	0.5303	19	35
3CHY	128	0.1209	0.5345	11	32
3CLA	213	0.0683	0.5297	20	39
3DFR	163	0.0867	0.5295	18	35
4BLM-A	256	0.0696	0.5163	17	45
5P21	166	0.0603	0.5155	12	35
1MSH	72	0.1839	0.5776	15	20
1UGO	99	0.1386	0.5708	15	19
2HIP	72	0.205515	0.503867	7	24
1BPL	189	0.104741	0.528325	13	33
1HDD	61	0.219144	0.5571	8	14
1I1B	153	0.103532	0.523601	15	35
1PGX	83	0.209796	0.505562	7	24
1SBT	275	0.065585	0.507135	15	37
1TAB(I)	82	0.289262	0.562105	8	8
1TIM	247	0.077003	0.50815	11	36
2C12*	439	0.040216	0.499146	13	129
2CAB	260	0.073242	0.508475	15	36
2HMQ	114	0.151123	0.53496	11	42
2PAB	127	0.148809	0.534725	12	33
2SOD	152	0.104557	0.523628	15	30
351C	82	0.171985	0.539269	10	33
3TMS	264	0.069846	0.507948	13	36
5CNA	237	0.078478	0.509583	10	47
1TAH*	318	0.042801	0.626127	15	48

Table 3.18: Length, Density, Clustering coefficient and Split-1 interaction values for Multi-Domain Proteins

PDB ID	Length	Density	Clustering Coefficient	Diameter	Split-1 Interactions
1BKS(B)*	255	0.0521	0.5036	10	97
1EZM*	299	0.0482	0.5131	15	68
1FNB*	296	0.0479	0.5077	15	93
1G6N*	200	0.0480	0.5276	11	53
1GPD	333	0.0522	0.5017	12	119
1LAP	481	0.0526	0.4964	11	272
1PFK	320	0.0569	0.5059	15	94
1RHD*	293	0.0518	0.5056	15	83
1SGT*	224	0.0373	0.5113	10	43
1VSG	362	0.0568	0.5096	12	144
2CYP*	297	0.0520	0.5091	15	88
3CD4*	178	0.0485	0.5314	10	37
3PGK	415	0.0476	0.5000	11	259
5FBP	313	0.0535	0.5075	15	94
8ADH	374	0.0507	0.5077	12	157
8ATC(A)	310	0.0492	0.5087	15	90
8ATC(B)*	146	0.0516	0.5436	9	30
1PHH	394	0.0606	0.5076	15	82
1ATN	372	0.0576	0.5113	16	105
3GRS	478	0.0596	0.5052	16	78
4GCR*	174	0.0568	0.529704	14	44
3PMG	561	0.0589	0.5050	16	96
8ACN	753	0.0614	0.5053	15	86
16PK	415	0.040603	0.492942	16	25
1CDG	688	0.023831	0.496012	16	34

1D5M*	181	0.087907	0.505141	11	48
1EPW	1290	0.011968	0.539807	46	108
1GPB	842	0.018117	0.571058	22	76
1HLE	345	0.046613	0.505769	21	72
1KF6*	243	0.064959	0.514333	19	58
1LAM	484	0.033674	0.498275	16	25
1LFI	691	0.023935	0.493405	19	113
1PGR*	215	0.076739	0.505445	15	53
1PHA	414	0.039668	0.501199	15	85
1PPR	312	0.050416	0.508619	23	61
1PRC	336	0.043764	0.500213	19	83
1SMP	471	0.03481	0.495285	16	52
1SMR	335	0.040848	0.62493	15	53
1SU4	994	0.015112	0.549449	37	63
1YGE	839	0.01781	0.569634	32	68
2BW4	340	0.043608	0.501682	21	36
2CPK	350	0.04865	0.493246	14	80
2LIV	344	0.047354	0.496132	15	80
5PEP	326	0.048404	0.508333	23	61
6ABP	306	0.053581	0.495881	14	82
5LDH	334	0.051799	0.509017	17	58
4APE	330	0.052268	0.509181	17	141
8TLN	316	0.053972	0.509817	17	135
9API	347	0.050957	0.507465	17	144

The dependence of these properties on the protein length is depicted in Figure 3.14 and a vertical line is drawn at 300 to see if these properties show any significant change in their values for single and multi-domain proteins. Since single domain proteins are expected to be more compact than multi-domain proteins, we first analyzed the dependence of the diameter of the graph, shown in Figure 3.14(a). It is seen to fluctuate about the average value 15.25 with no significant difference in its value between single and multi-domain proteins. That is, the diameter of a protein contact network is not a useful parameter for differentiating single domain proteins from multi-domain proteins.

The density of the proteins, computed using equation (2.6) is plotted in Figure 3.14(b). It may be noted that from the figure that small proteins, which are mainly single domain proteins, have higher values of density compared to multi-domain proteins. In other words, single domain proteins are more densely packed compared to multi-domain proteins. This is intuitively expected as multi-domain protein molecules exhibit large domain motions for their function. Using a cutoff of 0.06 (shown as a horizontal line), we find that most single domain proteins have a value < 0.06 except for five of them (1GOX, 2HAD, 2ACE, 1BKS, 2C12). Also all multi-domain proteins have density > 0.06 except for 5 proteins (1D5M, 1KF6, 1PGR, 1PHH, 8ACN). Thus, using only density as a parameter to distinguish between single and multi-domain proteins, we are able to correctly identify 90 out of a total of 100 proteins, i.e., an accuracy of 90%. Thus, density is a useful parameter for identifying single domain proteins.

In Figure 3.14(c) is plotted the clustering coefficient of all the proteins in the dataset. A gradual decrease in the value of the clustering coefficient (from 0.62 to 0.49) is seen with increase in the protein length. Using a cut-off value of 0.51, 9 single and 13 multi-domain proteins are respectively wrongly predicted. That is, a total of 22 wrong predictions out of 100. Thus, using only clustering coefficient as a parameter to distinguish between single and multi-domain proteins, we are able to correctly identify 78 out of a total of 100 proteins, i.e., an accuracy of 78%. Since most protein contact networks exhibit similar clustering coefficient values ~.53, with very smaller variations, we do not find it a useful parameter for identifying single domain proteins.

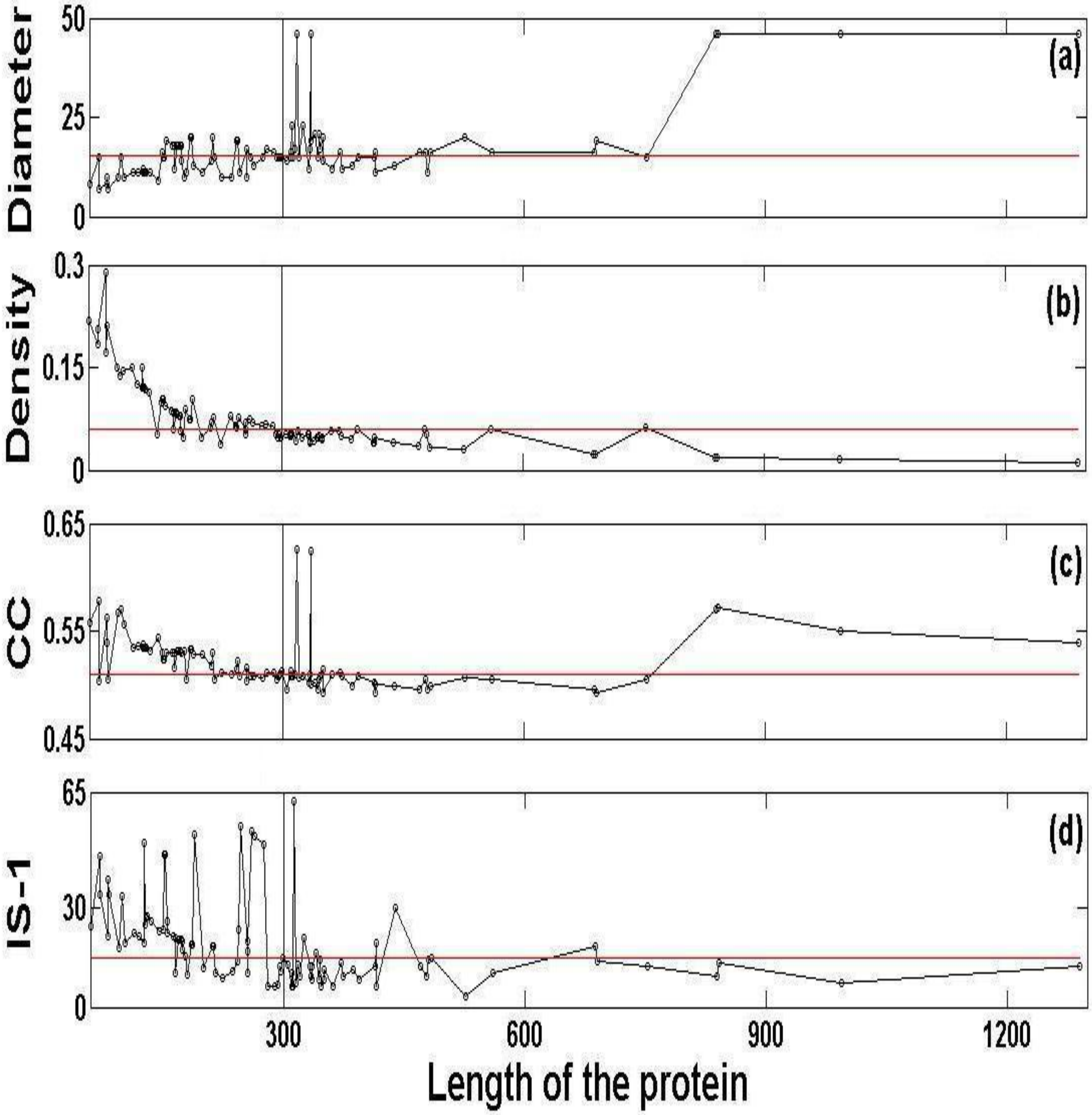


Figure 3.14 Graph properties for single and multi domain proteins.

If a single domain has been split into two by Newman's approach in the first phase of our approach, then a wrong prediction has been made. Hence we thought analyzing the interactions between the two clusters predicted by Newman's approach at the first step of its algorithm would provide a useful measure for distinguishing between single and multi-domain proteins. We refer to this as interaction strength at first split (IS-1) and is computed using equation (2.20). In Figure 3.14(d) is depicted the behaviour of the split-1 interaction strength as a function of the length of the protein for the 100 proteins. Since domains are independent and stable evolutionary units, the number of interactions within the domain is far more than between the domains. Hence we expect the interaction strength (IS-1) at split-1 to be higher for single domain proteins (since the two clusters in this case are suppose to be part of the same domain) compared to multi-domain proteins. In fact, we do observe a similar behaviour with most single domain proteins having a value > 15. Using this condition we are able to correctly identify 45 out of 50 single domain proteins, i.e., only five wrong predictions (1GOX, 1PYP, 2ACE, 1ULA, 2HAD). When analyzed on multi-domain proteins, we observed 44 out of 50 proteins having value ≤ 15, i.e., only six wrong predictions (16PK, 1CDG, 2BW4, 3CD4, 8ATC(B), 4APE). That is, a total of 89 correct predictions out of 100 is obtained only on the basis of this parameter. Here, we would like to point out that there is no particular reason for choosing 15 as the cut-off for the interaction strength, and changing the value of the threshold does affect the accuracy of the predictions. However, on analyzing the dataset of 100 proteins we observe that all most all single domain proteins exhibit large interactions, the few exceptions observed had typically lengths ~ 300.

Thus from the above analysis we observe that the graph density (≥ 0.06) and split-1 interaction strength (> 15) can be useful parameters for identifying single domain proteins. Since length is an important factor, we next analyzed the predictions on jointly using the condition (1) length and density cut-offs, and (2) length and IS-1 cut-offs. Using condition (1), that is, first filtering the proteins on the basis of their length and if its length ≤ 300 than classifying it as single domain protein only if its density ≥ 0.06. On applying this condition only seven wrong predictions were made and are listed in Table 3.19, giving an accuracy of 87.2%.

Table 3.19: False predictions for condition: length ≤ 300 and density ≥ 0.06.

PDB ID	Length	Density	CATH Annotation	Our Predictions
1GOX	350	0.0447	Single domain	Multi domain
2HAD	310	0.0521	Single domain	Multi domain
1KF6	243	0.0649	Multi domain	Single domain
1PGR	215	0.0767	Multi domain	Single domain
1D5M	181	0.0879	Multi domain	Single domain
2C12	439	0.0402	Single domain	Multi domain
1TAH	318	0.0428	Single domain	Multi domain

On using condition (2), that is, first filtering the proteins on the basis of their length and if its length ≤ 300 than classifying them as single domain proteins only if its interaction strength (IS-1) > 15, we observed only five false predictions, listed in the Table 3.20, giving an accuracy of 91%.

Table 3.20: False predictions for condition: length ≤ 300 and interaction strength, IS-1 > 15.

PDB ID	Length	Split-1 Interactions	CATH annotation	Our Predictions
2HAD	310	6.1	Single domain	Multi domain
3CD4	178	15.6	Multi domain	Single domain
8ATC(B)	146	22.8	Multi domain	Single domain
1GOX	350	8.2	Single domain	Multi domain
2ACE	527	3.2	Single domain	Multi domain

Since calculation of interaction strength after first split forms the first step in our approach for domain identification, and also it reliably distinguishes between single and multi-domain proteins, we have included it in our algorithm for domain identification and below we provide our prediction accuracies based on the condition of length and interaction strength, IS-1. Thus, once the protein has been predicted to contain more than one domain by these criteria, we proceed with our two-phase approach for identifying the number of domains and the domain boundaries.

3.3 Accuracy

Here we present our prediction accuracy on Jones *et al* dataset of 55 proteins. On using length filter cut-off (≤ 300) followed by split-1 interaction strength (> 15) for single domain proteins, we are able to correctly predict the number of domains in 50 out of 55 proteins, giving an accuracy of 91%. Our prediction accuracy results are summarized with other domain prediction tools, DOMAK, DomainParser and DDomain in Table 3.20. On the complete dataset of 100 proteins used in this analysis, correct predictions were made for 93 proteins, giving an accuracy of 93%.

Table 3.21: Prediction accuracies of various methods on Jones *et al.* dataset

Method	DOMAK	Domain Parser	DDomain	Our Approach
Accuracy	70%	83%	90%	91%

Chapter 4

4.1 Conclusion

Proteins are intrinsically modular, i.e., they are assembled from smaller structural modules that are spatially separable and can fold into independent compact 3-dimensional shapes known as *domains*. Domains can fold, function & evolve rather independently from the rest of the protein chain. Proteins evolve functionally due to the reorganizations of these domains (swapping and insertions). Thus various studies on proteins like its stability, structural transformations, folding and functional analysis typically begins with the decomposition of the protein structure into basic units, called structural domains. Databases such as SCOP and CATH have their first level of classification based on domains as this reduces a complex protein structure to a set of simpler yet structurally meaningful units, each of which can be analyzed independently. Hence the identification and analysis of domains forms the first step in understanding the protein functional and structural aspects. The specific characteristics of a domain that helps in its identification are compactness, existence of a hydrophobic core, low solvent accessibility, minimum size (~ 40) with single domain proteins being typically < 300 amino acids long and finally, large number of intra-domain interactions compared to inter-domain interactions. The domain prediction problem can be represented as a graph partitioning problem by constructing a protein contact network. As the graph partitioning algorithm divides the graph G into k disjoint partitions, the goal is to minimize the number of cuts in the edges between these partitions. This is in agreement with the typical characteristic feature of multi-domain proteins.

First step in domain identification problem is to identify whether the protein is single domain protein or it comprises of more than one domain. Once it is identified that the protein contains multiple domains, there are two problems - (i) to correctly identify the number of domains, and (ii) to accurately identify the domain boundaries. For a long time domains have been identified by visual inspection. However as the number of solved protein structures in PDB are increasing rapidly, there is an urgent need for the development of accurate methods for automatic domain identification. The unambiguous definition of a domain and the existence of noncontiguous domains add to the difficulty in developing an automated solution.

To address these issues we have proposed a simple method based on the analysis of graph spectral properties. The protein structure is represented by a graph by constructing a protein contact network, treating the backbone C_α atoms as nodes and an edge drawn between two C_α atoms if they are either connected by a peptide bond or are within a threshold distance of 7Å. The spectral analysis of a graph provides valuable information regarding global arrangement of nodes in the graph, important nodes in the graph, connectivity of each node and the clustering of nodes. This information on clustering of nodes is particularly useful in our domain identification approach.

We show here that the domain identification problem is very similar to community detection in a social network in the sense that the number of domains or communities is not known a priori and also both exhibit large number of connections within a community/domain than between communities/domains). This led us to implement Newman's modularity based community detection algorithm for domain identification.

First, we identify whether a given a protein is a single domain protein or consists of multiple domains. Various graph properties were analyzed for this purpose, *viz*., length, diameter, density, clustering coefficient and split-1 interaction strength (IS-1). We observe that the length and the interaction strength of the first split in Newman's modularity approach jointly provided a useful means for distinguishing single and multiple-domain proteins. This is done based on two criteria: first on a protein whose length is $\leq$ 300, the interaction strength between the clusters obtained after split-1 in Newman's approach is computed. If this interaction strength, IS-1 $\geq$ 15, then the protein is classified as a single domain protein and the program terminated. Else, the Newman's approach is continued. Here, both the top-down and bottom-up aspects of hierarchical clustering are used for dividing the protein structure into multiple domains. That is, starting from the whole protein structure we partition it iteratively into smaller compact structural units by Newman's community detection approach and then re-assemble them into domains. While decomposing the protein contact network into compact structural groups, the algorithm computes the quality of the decomposition at every step. It is defined as the difference in the number of edges falling within groups and the expected number in an equivalent network with edges placed at random and is referred to as modularity value of each split. The clusters obtained for the highest modularity value are then used to construct a coarse protein structure graph with each cluster treated as a node and weighted edges drawn between them based on the interactions between the clusters. The bottom-up approach is then implemented to agglomerate the nodes in this coarse graph by Clauset's optimization algorithm. The amalgamation step with highest modularity value identifies the structural domains in proteins. The details of this method are given in Chapter 2 and its implementation with few examples is discussed in Chapter 3.

To investigate the prediction accuracy of the proposed approach, the method has been implemented on a dataset of 100 proteins consisting of 50 single domain proteins and 50 multi-domain proteins (that includes both contiguous and non-contiguous domains). The dataset compiled by Jones *et al* that has been used by many domain identification algorithms is also used for the analysis. When compared with the annotations in CATH database our analysis on this dataset for distinguishing between single and multi-domain proteins results in 95 correct predictions (95% accuracy). Next, for the multi-domain proteins we compare our predictions with the annotations in CATH database and the results from a web-based domain prediction program, DomainParser. Out of the 50 multi-domain proteins, for 48 proteins the number of domains is correctly predicted giving an accuracy of 96%. The prediction is in good agreement not only in the number of domains predicted but also in the prediction of domain boundaries when compared to the annotations in manually curated CATH database. However, we do observe that for some multi domain proteins, when the domains are very close to each other in 3-dimensional space our approach may not be able to correctly identify the domain boundaries. Hence, human inspection is necessary, and should be used whenever the computational predictions appear doubtful. This point has been discussed in detail in Chapter 3 for protein 1CDG (Figure 3.7), in which two predicted domains had short intertwined regions and were merged into a single domain by visual inspection. Another example of incorrect prediction of number of domains is for protein 1YGE (Figure 3.12) which is reported to have 5 domains in the CATH database while our approach predicts only three domains. On analyzing the domain boundaries, we observed that two of the domain predictions matched with CATH annotation, but the third one comprised of three CATH annotated domains. The Newman's algorithm failed to split it further into three domains because of the close proximity of these domains in 3D space. In our analysis on 50 multi-domain proteins, we encountered only 2 incorrect predictions by our approach.

Non-contiguous domains are the result of genetic recombinations, domain swapping and insertions. Hence identifying their boundaries accurately is a major challenge. Unlike the graph spectral approach by Sistla *et.al,* which fails in predicting non-contiguous domains correctly, we have shown that our approach accurately predicts the domains and their boundaries in this case and results for a representative set of noncontiguous domains is summarized in Table 3.12. Other graph based methods like DOMAK and DomainParser also predict non-contiguous domains. The DOMAK approach scans the protein for the split position, i.e., for a domain with *N* segments, *2N* maximum split points need to be scanned, making it computationally expensive. DomainParser constructs a network model with each atom defined as a node and edges drawn if two atoms are within 4 Å distance. Compared to this, our protein network is constructed using each C_α atom as node thereby reducing the complexity of the network. Moreover DomainParser cannot predict single domains as the goal of the approach is to find best division of the network regardless of whether a good division even exists or not.

The Jones *et al* dataset of 55 proteins is commonly used by the domain identification community for testing the accuracy of their programs. When we tested our algorithm on this dataset we were able to correctly predict the number of domains in 50 proteins out of

55, giving an accuracy of 91%. When compared with other well-known domain prediction methods such as DOMAK, DomainParser, and DDomain with prediction accuracies of 70%, 83% and 90% respectively, we observed that the predictions of our approach are pretty good. The overall accuracy on the dataset of 100 proteins is 93%.

Thus the modularity based approach discussed here is a simple and elegant approach that is also computationally efficient. It distinguishes between single and multiple domains, correctly identifies number of multiple domains in a protein structure which may or may not be contiguous along the polypeptide chain, and requires no prior information on the number of domains.

Another important problem in domain identification is the analysis of protein repeat families wherein multiple copies of a single domain form regular arrangements of the repeating structural units providing extensive solvent accessible surface well suited to binding large substrates such as proteins and nucleic acids. In a recent work we show that the graph centrality measure betweenness can be used to identify the repeated domains and its boundaries (Ruchi *et al,* 2009). The analysis has been carried out on ARM and HEAT repeat families (results not included in the thesis).

Bibliography:

Ahmad, B., Kamal, M.Z. and Khan, R.H., (2004), Alkali-induced Conformational Transition in Different Domains of Bovine Serum Albumin, Protein Peptide Letters, 11, 307-315.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J., (1997), Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs, Nucleic Acids Research, 25, 3389-3402.

Bairoch, A., and Apweiler, R., (1997), The SWISS-PROT protein sequence data bank and its supplement TrEMBL, Nucleic Acids research, 25, 31-36.

Barabasi, Albert-Laszlo and Albert, Reka, (1999), Emergence of Scaling in Random Networks, Science, 286, 509-512.

Bateman, A., Birney, E., Durbin, R., Eddy, S.E., Lowe, K.L. and Sonnhammer, E.L., (2000), The Pfam Protein Families Database, Nucleic Acids Research, 28, 263-266.

Bennett, M.J., Schlunegger, M.P., Eisenberg, D., (1995), 3D Domain Swapping: A Mechanism for Oligomer Assembly, Protein Science, 4, 12, 2455-2468.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E., (2000), The Protein Data Bank, Nucleic Acids Research, 28, 235-242.

Boeckmann, B., Bairoch, A., Apweiler, R., (2003), The SWISS-PROT Protein Knowledgebase and its Supplement TrEMBL, Nucleic Acids Research, 31(1), 365-370.

Bork P., (1991), Shuffled Domains in Extracellular Proteins, FEBS Lett, 286 (1-2), 47-54.

Chen, J., Wang, J. and Wang, W., 2004, Transition States for Folding of Circular-Permuted Proteins, Proteins, 57, 153-171.

Chenal, A., Nizard, P., Forge, V., Pugniere, M., Roy, M.O., Mani, J.C., Guillain, F. and Gillet, D., 2002, Does Fusion of Domains from Unrelated Proteins Affect their Folding Pathways and the Structural Changes Involved in their Function? A case study with the diphtheria toxin T domain, Protein Engineering, 15,383-391.

Cheng, J., Sweredoski, M. J. and Baldi, P., (2006), DOMpro: Protein Domain Prediction Using Profiles, Secondary Structure, Relative Solvent Accessibility, and Recursive Neural Networks, Data Min. Knowledge Discovery, 13, 1-10.

Clauset, A., Newman, M. E. J. and Cristopher Moore, (2004), Finding Community Structure in Very Large Networks, Physical Review, 70 id. 066111.

Coleman., Thomas, F.; Moré,, Jorge, J., (1983), Estimation of Sparse Jacobian Matrices and Graph Coloring Problems, SIAM Journal on Numerical Analysis, 20 (1), 187-209.

Conte, L.L., Ailey, B., Hubbard, T.J.P., Brenner, S.E., Murzin, A.G., Chothia, C., (2000), A Structural Classification of Protein Database, Nucleic Acid Research, 28, 257-259.

Corpet, F., Servant, F., Gouzy, J., Kahn, D., (2000), ProDom and ProDom-CG: Tools for Protein Domain Analysis and Whole Genome Comparisons, Nucleic Acids Research, 28, 267-269.

Crippen, G. M., (1978), The Tree Structural Organization of Proteins, Journal of Molecular Biology, 126, 315-332.

Csardi, G. and Nepusz, T., (2006), The igraph Software Package for Complex Network Research, International Journal of Complex Systems,1695.

Cunningham, E. L. and Agard, D.A., (2003), Interdependent Folding of the N- and C-terminal Domains Defines the Cooperative Folding of α-lytic Protease, Biochemistry, 42, 13212-13219.

Doolittle, R. F., (1995), The Multiplicity of Domains in Proteins, Annu Rev Biochem, 64, 287-314.

Elsner, U., (1997), Graph Partitioning: A Survey (Technische Universität Chemnitz, Chemnitz, Germany) Technical Report, 97-27.

Erdos, P. and Renyi, A., (1959), On Random Graphs I, Publicationes Mathematicae, 6, 290-297.

Felsenstein, J., (1985), Confidence-limits on Phylogenies- An approach using the Bootstrap, Evolution, 39, 783-791.

Fjallstromas, P. (1998), Linkoping Electronic Articles in Computer and Information Science, 3, 10.

Flake, G. W., Lawrence, S. R., Giles, C. L. and Coetzee F. M., (2002), IEEE Computer, 35,66-71.

Ford, L. R.; Fulkerson, D. R., (1956), Maximal Flow through a Network, Canadian Journal of Mathematics, 8, 399-404.

Freeman, W.H., and Co Stryer, (2002), Biochemistry, Fifth Edition.

Ganesh, B. and Somdatta, S., (2007), Network Properties of Protein Structures, Physica, 346, 27-33.

Garel, J., (1992), Folding of Large Proteins: Multidomain and multisubunit Proteins, In Creighton, T., editor, Protein Folding, 405-454.

Garey, M., (1976), Some Simplified NP-complete Graph Problems, Theoretical Computer science, 1,237-267.

George, R.A., Lin, K. and Heringa, J., (2005), Scooby-Domain: Prediction of Globular Domains in Protein Sequence, Nucleic Acids Research, 33 (Web Server issue), W160-W163.

Ghelis, C., Yon, J.M., (1979), Conformational Coupling between Structural Units. A Decisive Step in the Functional Structure Formation, C R Seances Acad Sci D, 289 (2), 197-199.

Girvan, M., Newman, M. E. J., (2002), Proceedings of National Academy of Science, USA, 99,7821-826.

Go, M., (1983), Modular structural units, exons and function in chicken lysozyme, Proceedings of National Academy of Science, USA, 80,1964-1968

Gould, P., (1967), The Geographical Interpretation of Eigenvalues, Transactions of Institute of British Geographers,42,53-58.

Gouzy, J., Corpet, F. and Kahn, D., (1999), Whole genome protein domain analysis using a new method for domain clustering, Computational Chemistry, 23, 333-340.

Greene, L., Higman, V., (2003), Uncovering Network Systems within Protein Structures, Journal of Molecular Biology, 334, 781-791.

Grigoriev, I., Mironov, A. and Rakhmaninova, A., (1994), Inter-helical Contacts Determining the Architecture of Alpha-helical Globular Proteins, Journal of Biomolecular Structure and Dynamics, 12, 559-572.

Guimera, R.,(2006), The Real Communication Network Behind the formal chart: Community structure in organizations, Journal of Economic Behavior & Organization, Vol. 61, 653-667.

Hari Krishna Yalamanchili and Nita Parekh, (2009), Graph Spectral Approach for Identifying Protein Domains, LNCS, Proceedings of International Conference on Bioinformatics and Computational Biology, BICoB 2009, 437-448.

Heringa, J., Taylor, WR., (1997), Three-dimensional Domain Duplication, Swapping and Stealing, Current Opinion in Structural Biology, 7 (3), 416-421.

Holm, L., Sander, C., (1994), Parser for Protein Folding Units, Proteins, 19, 256-268.

Hongyi Zhou et al, (2007), DDOMAIN: DDOMAIN: Dividing Structures into Domains using a Normalized Domain-Domain Interaction Profile, Protein Sciece, 16(5), 947-955.

Islam, S.A., Luo, J., Sternberg, M.J., (1995), Identification and Analysis of Domains in Proteins, Protein Engineering, 8 (6), 513-25.

Janin, J. and Wodak, S. J., (1983), Structural Domains in Proteins and their role in the Dynamics of Protein Function, Progress in Biophysics and Molecular Biology, 42, 21-78.

Jones, S., Stewart, M., Michie, A., Swindells, M.B., Orengo, C., Thorton, J.M., (1998), Domain Assignment for Protein Structures using a Consensus Approach, Characterization and Analysis, Protein Science, 7, 233-242.

Jorg Schultz, Frank Milpetz, Peer Bork and Chris P. Ponting, (1998), SMART: A Simple Modular Architecture Research Tool: Identification of Signaling Domains, Proceedings of National Academy of Science, USA, 95,11, 5857-5864.

Kabsch, W., Sander, C., (1983), Dictionary of Protein Secondary Structure-Pattern-recognition of Hydrogen-bonded and Geometrical Features, Biopolymers, 22,2577-2637.

Kannan and Vishveshwara, (1999), Identification of Side-chain Clusters in Protein Structures by Graph Spectral Method, Journal of Molecular Biology, 292(2), 441-464.

Karisch, S.E. et al, (1997), Solving Graph Bisection Problems with Semidefinite Programming, Technical Report, DIKU-TR97/9.

Kernighan, B. W., Lin, Shen. (1970), An Efficient Heuristic Procedure for Partitioning Graphs, Bell Systems Technical Journal, 49, 291-307.

Koch, I., Kaden, F. and Selbig, J., (1992), Analysis of Protein Sheet Topologies by Graph-theoretical methods, Protein Structure Function and Genetics, 12, 314-323.

L Da Costa et al, (2006), Characterization of Complex Networks: A survey of Measurements, arXiv, 424,175-308.

Leon Danon et al, (2005), Comparing Community Structure Identification, Journal of Statistical Mechanics, P09008.

Liisa Holm and Chris Sander, (1994), Parser for Protein Folding Units, PROTEINS: Structure, Function, and Genetics, 19,256-268.

Marchler-Bauer, A., Anderson, J.B., Chitsaz, F., Derbyshire, M.K., DeWeese-Scott, C., Fong, J.H., Geer, L.Y., Geer, R.C., Gonzales, N.R., Gwadz, M., He, S., Hurwitz, D.I., Jackson, J.D., Ke, Z., Lanczycki, C.J., Liebert, C.A., Liu, C., Lu, F., Lu, S., Marchler, G.H., Mullokandov, M., Song, J.S., Tasneem, A., Thanki, N., Yamashita, R.A., Zhang, D., Zhang, N., Bryant, S.H., (2005), CDD: CDD: a Conserved Domain Database for Protein Classification, Nucleic Acids Research, 33, 192-196.

Marchler-Bauer, A., Anderson, J.B., DeWeese-Scott, C., Fedorova, N.D., Geer, L.Y., He, S., Hurwitz, D.I., Jackson, J.D., Jacobs, A.R., Lanczycki, C.J., Liebert, C.A., Liu, C., Madej, T., Marchler, G.H., Mazumder, R., Nikolskaya, A.N., Panchenko, A.R., Rao, B.S., Shoemaker, B.A., Simonyan, V., Song, J.S., Thiessen, P.A., Vasudevan, S., Wang, Y., Yamashita, R.A., Yin, J.J. and Bryant, S.H., (2003), CDD: A Curated Entrez Database of Conserved Domain Alignments, Nucleic Acids Research, 31(1),383-387.

Marsden, R.L., McGuffin, L.J. and Jones, D.T., (2002), Rapid Protein Domain Assignment from Amino acid Sequence using predicted secondary structure, Protein Science, 11, 2814-2824.

Newman, M. E. J., (2004), European Physical Journal B, 38,321-330.

Newman, M.E.J., (2006), Finding Community Structure in Networks using the Eigenvectors of

Matrices, Physical Review, 74, id. 036104.

Newman, M.E.J., (2006), Modularity and Community Structure in Networks, Proceedings of the National Academy of Sciences, USA, 103, 8577-8582.

Nishikawa, K., Ooi, T., Isogai, Y., Saito, N., (1972), Tertiary Structure of Proteins I. Representation and computation of the conformation, Journal of the Physical Society of Japan, 32, 1331-1337.

Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., Thornton, J.M., (1997), CATH--A Hierarchic Classification of Protein Domain Structures, Structure, 1093-1108.

Ostermeier, M., (2005), Engineering Allosteric Protein Switches by Domain Insertion, Protein Engineering, Design and Selection, 18, 359-364.

Ostermeier, M., Benkovic, S.J., (2000), Evolution of Protein Function by Domain Swapping. Advanced Protein Chemistry, 55, 29-77.

Patra, S., Vishveshwara, S., (2000), Backbone Cluster Identification in Proteins by a Graph Theoretical Method, Biophysical Chemistry, 84, 13-25.

Per-Olof, F., (1998), Algorithms for Graph Partitioning: A Survey. Linkoping Electronic Articles in Computer and Information Science, 3, 10.

Phillips, D.C., (1970), British Biochemistry, Pasrandpresent,. London: Academic Press, 11-28.

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

Rashin, A.A., (1981), Location of Domains in Globular Proteins, Nature, 291,85-86.

Redfern, O.C., Harrison, A., Dallman, T., Pearl, F.M., Orengo, C.A., (2007), CATHEDRAL: A Fast and Effective Algorithm to Predict Folds and Domain Boundaries from Multidomain Protein Structures, PLoS Computational Biology, 11, 232.

Richardson, J. S., (1981), The Anatomy and Taxonomy of Protein Structure, Advanced Protein Chemistry, 34, 167-339.

Rose, G. D., (1979), Hierarchic Organization of Domains in Globular Proteins, Journal of Molecular Biology, 134, 447-470.

Rossman, M. G. and Liljas, A., (1974), Letter: Recognition of Structural Domains in Globular Proteins., Journal of Molecular Biology, 85, 177-181.

Ruchi Jain, Hari Krishna Yalamanchili and Nita Parekh, (2009), Identifying Structural Repeats in Protein using Graph Centrality Measures, Proceedings of World Congress on Nature and Biologically Inspired Computing.

Russell, R. B., (1994), Domain Insertion, Protein Engineering, 7, 1407-1410.

Saraswathi Vishveshwara et al, (2002), Protein Structure Insights from graph theory. Journal of Theoretical and Computational Chemistry, 1(1), 187-211.

Savageau, M.A., (1986), Proteins of Escherichia coli come in sizes that are multiples of 14 kDa: Domain Concepts and Evolutionary Implications, Proceedings of the National Academy of Sciences, U S A, 83 (5), 1198-1202.

Siddiqui, A.S., Barton, G.J., (1995), Continuous and discontinuous domains, an algorithm for the automatic generation of reliable protein domain definitions. Protein Science, 4, 872--884

Sistla, Brinda, Ramesh, K., K.V. and Saraswathi Vishveshwara, (2005), Identification of Domains and Domain Interface Residues in Multidomain Proteins from Graph Spectral Method. Protein Structure, Function, and Bioinformatics, 59, 616-626.

Sonnhammer, S.R., Eddy. and Durbin, R., (1997), Pfam: A Comprehensive Database of Protein Families based on seed Alignments, Proteins, 28, 405-420.

Sowdhamini, R., Blundell, T.L., (1995), An Automatic Method Involving Cluster Analysis of Secondary Structures for the Identification of Domains in Proteins, Protein Science, 4 (3), 506-520.

Stella Veretnik and Ilya Shindyalov, (2007), Computational Methods for Domain Partitioning of Protein Structures. Chapter 4, Computational Methods for Protein Structure Prediction and Modeling, 125-145.

Swindells, M. B., (1995), A Procedure for Detecting Structural Domains in Proteins, Protein Science, 4, 103-112.

Swindells, M.B., (1995), A procedure for the automatic determination of hydrophobic cores in protein structures. Protein Science, 4, 93-102.

Tatusov RL, Koonin EV, Lipman DJ., (1997), COG: Clusters of orthologous groups - a genomic perspective on protein families. Science, 278(5338),631-7.

Tsai, C.J. and Nussinov R., (1997), Hydrophobic folding units derived from dissimilar monomer structures and their interactions, Protein Science, 6 (1), 24-42.

Wasserman, S., Faust, K., (1994), Social Network Analysis (Cambridge Univ. Press, Cambridge, U.K.),

Watts, Duncan J.; Strogatz, Steven H., (1998), Collective Dynamics of Small-World Networks. Nature, 393, 440-442.

Wetlaufer, D.B., (1973), Nucleation: Rapid Folding, and Globular Intrachain Regions in proteins. Proceedings of the National Academy of Sciences, U S A, 70 (3), 697-701.

White, H. C., Boorman, S. A., Breiger, R. L., (1976), American Journal of Sociology, 81,730-779.

Xu, Y., Xu, D., Gabow, H.N., (2000), Protein Domain Decomposition using a Graph-theoretic approach. Bioinformatics, 16, 1091-1104.

Zehfus, M.H., Rose, G.D., (1986), Compact Units in Proteins, Biochemistry, 25, 5759-5765.

Zhou, Y., Vitkup, D. and Karplus, M., (1999), Native Proteins are Surface-molten Solids: Application of the Lindemann Criterion for the Solid versus Liquid state, Journal of Molecular Biology, 285, 1371-1375.