Developing an automated method to identify structural domains in proteins

Thesis submitted in partial fulfilment
of the requirements for the degree of

Master of Science (by Research)
in
Computational Natural Sciences

by

Anirudh Tiwari
201164104
anirudh.tiwari@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)
Hyderabad - 500 032, India
November 2016

To my family.

International Institute of Information Technology
Hyderabad, India

DECLARATION OF AUTHORSHIP

I, Anirudh Tiwari, declare that the thesis titled ”Developing an automated method to identify structural domains in proteins” and work presented in it are my own. I confirm that this work was done wholly or mainly while in candidature for a research degree at this university.

Date

Signature of the candidate

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled ”Developing an automated method to identify structural domains in proteins” by Anirudh Tiwari, has been carried out under my supervision and is not submitted elsewhere for a degree.

____

Date

____

Advisor: Dr. Nita Parekh

Acknowledgements

Abstract

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 An introduction to proteins

Proteins are complex organic compounds formed out of long chains of amino acids which are connected by peptide bonds. They are one of the most essential class of bio-molecules and are the building blocks of our body. Proteins are biologically synthesized by assembling of amino acids using information encoded in genes and behave according to the respective gene from which they were synthesized. Proteins play a wide variety of roles and perform many functions which control the growth, immunity and strength of our body. For example, enzymes are those proteins which catalyzes many biochemical reactions and thus are essential to metabolism. Enzymes also play a vital role in DNA replication, DNA repair and transcription. Other functions of proteins include cell signalling, immune response, cell cycle etc. The reason behind proteins performing such a wide variety of functions is their ability to tightly bind to other molecules in a specific fashion. For example, antibodies are protein components of an adaptive immune system which binds to foreign substances in the body and protect them from destruction. Since their binding ability allows them to be functionally active, it becomes essential to study their 3-D structure to understand this behaviour. Protein structure is organized in four hierarchical levels: Primary, Secondary, Tertiary and Quaternary as shown in figure 1.1.

Primary: The linear sequence of amino acid residues is a protein's primary structure. By convention, the primary structure begins at N-terminal(Amino-terminal) and ends at the C-terminal(Carboxyl-terminal). This is a 1-D structure and is determined by the gene from which it is synthesized. Although the primary structure plays a great deal in determining the tertiary structure(the actual 3-D structure) of a protein, but it can't be deduced just by simply observing the sequence.

Secondary: Secondary structure of a protein is its general 3-D form of local segments of protein. Formally, it is defined by the pattern of Hydrogen bonds between Amine Hydrogen and Carboxyl Oxygen atoms contained in the backbone residues. There are broadly three types of secondary structures:

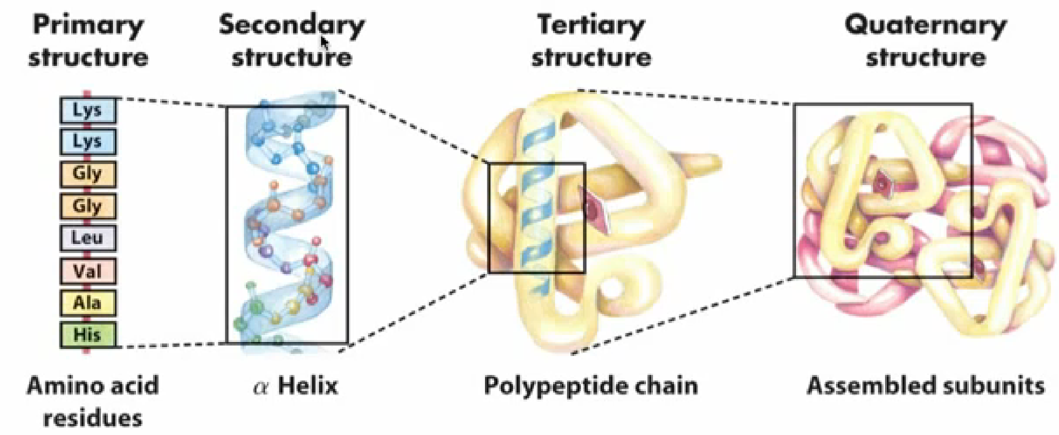

Figure 1.1: Overview of structural hierarchy of proteins.

Alpha Helices: This is one of the most common type of secondary structure and has a coiled spiral type of formation in which there is a Hydrogen between the N-H group of every backbone residue with the third/fourth residue's C=O group.

Beta Sheets: It is another commonly observed secondary structure which consists of laterally interconnected Beta strands by at least 2-3 Hydrogen bonds. A Beta strand by definition is a peptide chain of around 3-10 amino acids. The sheets can be either parallel or antiparallel, if the N-terminus of the adjacent strands are aligned then the sheets are parallel else they are antiparallel.

Turns: Turns essentially play the role of a bridge where the polypeptide chain changes its overall direction. They are typically around 4-5 residue in length and are not a part of either alpha helices or beta sheets.

Tertiary: The tertiary structure of a protein essentially signifies its overall geometric shape. It encompasses a single backbone chain and can have multiple secondary structures. The tertiary structure describes the relative positioning and packing of alpha helices and beta sheets and it can be studied by reading the atomic coordinates of the backbone residues.

Quaternary: Quaternary structure is the arrangement of multiple protein subunits. These subunits are not usually covalently connected, but might have disulphide bonds between them. Many proteins exist as assemblies of multiple polypeptide chains. Some examples of quaternary structures include hemoglobin, DNA polymerase etc. Complexes of two or more polypeptides are called multimers. More specifically, an assembly of two will be called a dimer, of three will be called a trimer and so on.

Motifs: A structural motif describes the connectivity between secondary structure elements. A motif can have very few elements like helix-turn-helix which has 3 elements or can be complex like Greek key which consists of four antiparallel beta strands and their linking loops(figure 1.2).

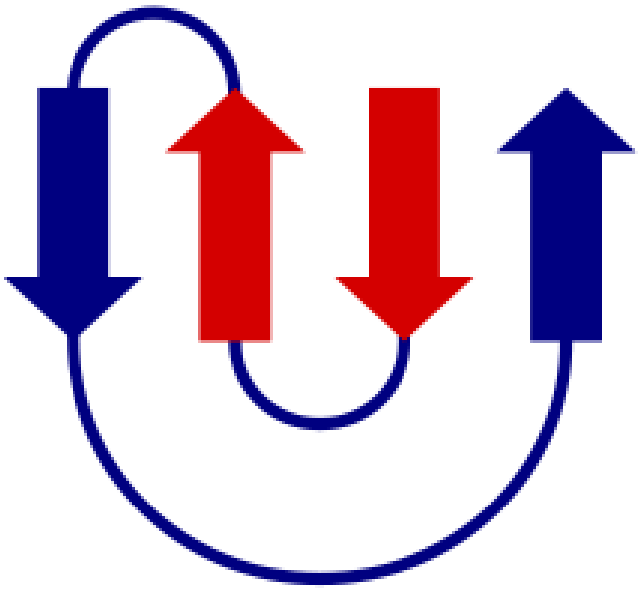

Figure 1.2: A Greek key motif showing 4 anti-parallel beta strands with their links.

Domains: A domain is a conserved and compact region of a protein which can evolve and function independently of the rest of the protein. Thus, a domain can be considered as the very basic functional unit in a protein. Moreover, a domain has a hydrophobic core and is independently stable. A protein can have multiple domains and a domain can appear in many different proteins imparting different functionality with every different combination.

1.2 The importance of domain identification

Research on the topic of domains, which spans over 40 years, established the following properties it possess, they are (1) stability; (2) compactness; (3) ability to fold autonomously; (4) possessing a hydrophobic core; (5) conserved across evolution; (6) performing a specific function[1]. The number of distinct domains are currently way over the thousand mark and are represented by unique fold in SCOP[2] classification or by unique topologies in CATH[3] classification. The length of a domain varies, it can be as small as 30 residues long and can be even larger than 300 residues. Large domains generally possess multiple hydrophobic

cores. A protein chain can have multiple domains imparting multi-functionality. In a multi-domain protein, each domain can perform different functions in association with the other domains or can function independently. Different proteins containing the same domain content may also have different functions because of the difference in the order of arrangement of the domains. The various arrangements/rearrangements of domains is due to genetic recombination which is also responsible for domain swapping and insertion.
Domain swapping: 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: Non-contiguous domains are formed as a result of domain insertion. A non-contiguous domain is one in which its segments are spatially close to each other but on a sequence level have another domain in-between them as shown in figure 1.3

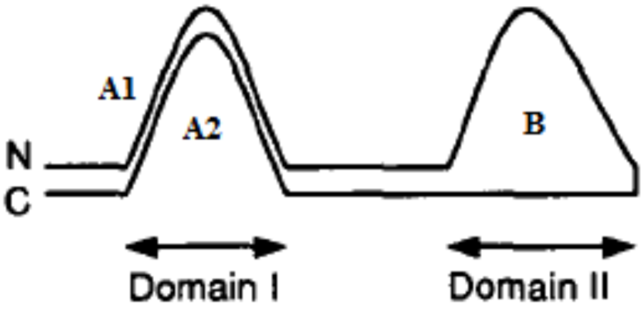

Figure 1.3: The discontiguous domain(A) has two segments A1 and A2, with domain B inserted between them.

The functional aspects of domains imposes their vitality in all the living organisms and thus their identification becomes an important task. But with all the various properties a domain exhibit, there has not been a consensus amongst the researchers over its concrete definition. Furthermore, varying lengths and non-contiguous nature of domains makes their identification an even more complex problem. Although the boundaries of a domain can be identified by visual inspection, it becomes an uphill task to regularly update the domain database especially when the number of resolved protein structures are increasing exponentially with time. This calls for developing automated methods which can identify domain boundaries just by feeding on a protein sequence or structure or a combination of both. This is not an easy task by any means and a lot of research has already been done to tackle it. While many approaches have produced successful results, but even the best ones lack the robustness required to consistently perform over the entire database of resolved structures and there is a long way to go.

1.3 Literature Survey

The identification of domains was initially done manually by human experts by examining the the protein structures. The first reported study is by Wetlaufer in 1973[4] by constructing three-dimensional peptide chain models which was facilitated by the then available x-ray structures and stereoscopic images. He observed distinct structural "regions" which were contiguous as well as non-contiguous and defined these "regions" as a section of peptide chain that can be enclosed in a compact volume. This laid down the foundation of structural domains and their identification, which has been a topic of scientific research for over 40 years now.

While manual curation is an accurate and reliable approach, what it lacks is consistency and speed. With an observed increase in the number of solved structures, automated methods have been proposed for domain identification. These methods are fast, systematic and takes into account various properties possessed by structural domains. A myriad of algorithms have been proposed with each having it's own pros and cons, a common observation is that almost all of them employ a hierarchical approach. An extensive review of domain identification algorithms is given by Stella et al[5]. In the following sections we briefly discuss some of the important domain identification approaches proposed till date.

1.3.1 Sequence based methods

Predicting domains based on the sequence information only and without the knowledge of their 3-D structure is a tough task. Some of the early approaches to the problem ranged from assembling secondary structure elements into domains to identifying domains as those areas having high residue contact density[6]. A more recent approach guesses the number of domains based on their size based distribution[7]. Unfortunately, these methods had poor results and were unreliable.

Current methods rely more on the fact that a domain is a continuous sequence of amino acids that recurs in the protein space. Thus, domains are evolutionary in nature and are those segments of protein that are conserved and reused throughout evolution. Hence, it is observed that sequences which have a substantial sequence similarity(>30%) share common domains that possesses a common fold and thus usually share similarity in function[8]. Such methods employ an alignment approach where domains are identified by aligning the target sequence against sequences present in the database with known boundaries[9]. This method is efficient but relies on the existence of homology. Also, this method fails in identifying non-contiguous domains as it assigns each conserved segment to a separate domain. In addition to sequence alignments, some methods employ machine learning to further enhance their prediction.

Some of the methods using the above techniques are discussed below.

CHOPnet

CHOPnet[10] is a de novo method that predicts structural domains in the absence of homology to known domains. The method is based on neural networks and relies exclusively on information available for all proteins. Multiple sequence alignments are obtained by searching with PSI-BLAST[11] against all known sequences contained in SWISS-PROT, TrEMBL[12] & PDB[13]. All hits below a PSI-BLAST E-value of 10^{-3} are filtered and included in the sequence profile. For these filtered alignments, amino acid composition, predicted secondary structure and solvent accessibility are calculated and used as an input to the neural network. The aforementioned neural network is a three layer feed forward artificial neural network using the standard back-propagation algorithm[14]. The neural network takes 57 input nodes from consecutive sequences of 5 residues(shifted through the protein). For each residue in the local sequence window, three nodes encoded secondary structure representing helix, strand & loop using output from PROFsec, one node representing solvent accessibility from PROFacc and one another node representing the sequence conservation using HSSP files[15]. Further nodes were added to incorporate the features of the entire 5 residue long segment, (a) difference in the secondary structure content between the flanking regions of the segment[8 nodes]; (b) difference in solvent accessibility[4 nodes]; (c) position of sequence segment with respect to N- and C-termini[8 nodes]; (d) the flexibility index[16][2 nodes]; (e) and the amino acid composition in the profile for the entire window[6 nodes]. Lastly, 4 global nodes were added to represent the length of the protein. The hidden layer had 3 nodes and the output layer had 2 nodes, one denoting the domain boundaries and the other one denoting the not-domain boundaries.

Domcut

Domcut[17] is a program which predicts inter-domain linker regions based on the difference in amino acid composition between domain and linker regions. Only those sequences which have a length in range 50-500 are considered as domains. On the other hand, a linker region must be: (1) connecting two adjacent domains defined above; (2) in the range from 10 to 100 residues; (3) and not containing membrane spanning regions. To represent the preference of amino acid residues in linker regions, the linker index S_i for amino acid residue i is calculated as shown in equation 1.1:

$$S_i = -ln(f_i^{linker}/f_i^{domain}) \tag{1.1}$$

Where $f_i^{linker}/f_i^{domain}$ is the frequency of amino acid residue i in the linker/domain region respectively. The negative value of S_i means the residue preferably exists in the linker region. Then, a linker preference profile is constructed by plotting average linker index values along an amino acid sequence using a sliding window. The linker regions correspond to the troughs in the profile as shown in figure 1.4.

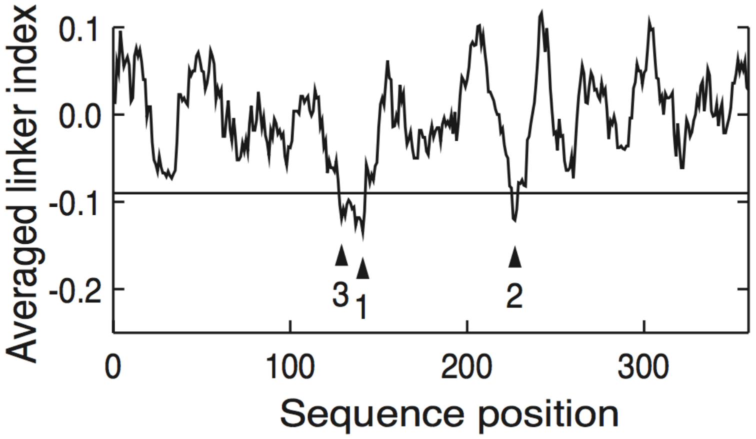

Figure 1.4: An example of linker preference profile generated by DomCut

FIEFDom

FIEFDom[18](Fuzzy Integration of Extracted Fragments for Domains) is a method to predict domain boundaries of proteins from a given sequence profile using Fuzzy nearest neighbour algorithm[19]. A three step procedure is used to predict the boundaries. First, a Position Specific Scoring Matrix (PSSM) for the query sequence is generated using PSI-BLAST program and searching against a non-redundant protein sequence database containing information about domain boundaries, e.g., CATH & SCOP. PSSM is a 2-D matrix which represents the likelihood of each amino acid occurring at every position along the protein sequence. The generated profile is then used to search for similar fragments in the database by doing profile-sequence alignment between the query profile and the proteins in the database using PSI-BLAST program. The expectation value (e-value) is set to 10,000 in this step to ensure that both large and small sized fragments are retrieved. These alignments obtained are parsed and scored using formula 1.2:

$$S = max\{1, 7 + log_{10}(e\text{-}value)\} \tag{1.2}$$

where S is a dissimilarity measure. Thus, the sequence fragments in the database that have high sequence similarity and high statistical significance (or low e-value) with the sub-sequences of the query protein have low scores. Finally, the domain boundaries (if any) are predicted using the scored fragments. For each residue, the P_B is calculated from the domain boundary memberships (B) of the residues in the fragments that are aligned with the current residue. The P_B of the query protein is calculated using the following expression(equation 1.3) for the Fuzzy

Nearest Neighbour algorithm:

$$P_B(r) = \frac{\sum_{j=1}^{K} B_j(r)(1/S_j^{2/(m-1)})}{\sum_{j=1}^{K}(1/S_j^{2/(m-1)})} \tag{1.3}$$

where, r is the current residue identifier, K is the number of fragments that have a residue aligned with the current residue r, $\mathrm{B}_j(\mathrm{r})\epsilon$(0 if the residue lies in the domain and 1 if the residue lies on the domain boundary) is the domain boundary membership of the residue in the j^{th} fragment that has a residue aligned with the current residue r, S_j is the score for the j^{th} fragment defined in the first equation, and m is a fuzzifier that controls the weight of the dissimilarity measure, S.

EVEREST

EVEREST[20](EVolutionary Ensembles of REcurrent SegmenTs) is a domain family identification method which assumes that a domain is a continuous sequence of amino acids that recurs(non trivially) in the protein space. Based on this working principle, EVEREST uses a rigorous process to identify domain families. It begins by constructing a library of non-redundant protein segments that emerge in all vs. all pairwise sequence alignment using BLAST. Each of these protein segment act as a representative of all those proteins which has a similarity score less than 1e - 90. Next, it was found that each of these protein segments had repeated regions and the presence of three or more of these regions can lead to false conclusions. Thus, to discover such repeating regions, each segment is compared to itself in an iterative fashion by using Smith-Waterman sequence comparison algorithm[21] such that positions that were matched in previous iterations were not matched again. After the internal removal of repeated regions, BLAST is again run on them to find segments which are similar but with a very relaxed similarity criteria(E-score $<$ 100). Next, it cluster these segments into putative domain families by using the average linkage clustering algorithm. A HMM profile is then constructed for each of the putative families, and the procedure is iterated by using the HMM profiles to recreate the putative domains database. This procedure is repeated three times and the domain families defined by the third iteration HMM are clustered into sets of overlapping families. Final domains are defined by voting of all the HMMs of each set.

1.3.2 Structure based methods

Systematic identification of structural domains by analyzing their 3-D structures has been going on for over 40 years now, the initial strides being made by Phillips(1970)[22], Ooi & Nishikawa(1973)[23]. Both of them constructed the Cα-Cα distance plots and identified domains by visually inspecting for triangular regions near the diagonal as shown in figure 1.5

Rossman & Liljas(1974)[24] performed the first systematic algorithmic analysis of Cα-Cα distance plots and have been followed by myriads of algorithms trying

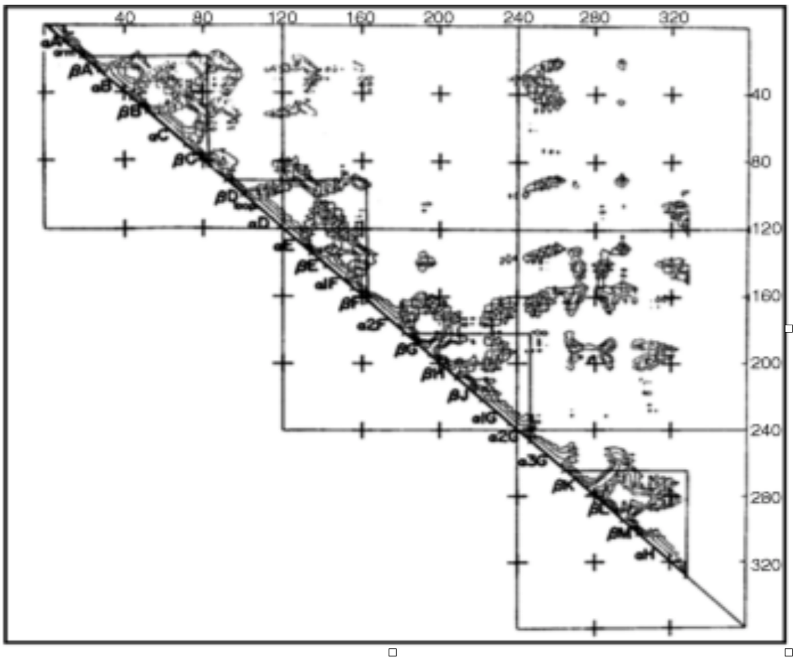

Figure 1.5: Cα-Cα distance plot of lactase dehydrogenase

to identify structural domains. Though differing in their implementations and peculiarities, almost all of these algorithms employ a common technique of decomposing the protein structure in a hierarchical fashion. On top of that, some of the methods work on the assumption that intra-domain residual contact density would be higher than that of inter-domain while others make use of the fact that a domain is a compact structure with a hydrophobic core. These are just some of the basic methods that are used and some of the prominent publications which have used these techniques are discussed below.

DETECTIVE

DETECTIVE[25] is a tool developed by Mark B. Swindells which identifies structural domains based on the assumption that each domain will be constituted of a hydrophobic core except for extremely small proteins which are held together by numerous disulfide bridges. In another related work, he has described a conceptually simple and computationally efficient algorithm for identifying hydrophobic cores in proteins[26]. According to him, a hydrophobic core is a collection of residue sites with low solvent accessibility, which are located in regions of regular secondary structure and whose non-polar side-chain moieties interact with one another. This definition serves as the working premise for DETECTIVE and the steps for domain identification are as follows. (1)Hydrophobic cores are identified with 7% solvent accessibility[27] cutoff; (2) Then, Isolated cores and cores having less than 5 sites are removed; (3) Further, sequentially adjacent cores are merged together to initiate the domain assignment process; (4) Next, these initial domains are extended to the entire protein by merging the initially unassigned sites to that domain with which it has the highest number of contacts. Again, isolated sites are removed and adjacent sites are merged in a similar manner as in steps 2 & 3; (5) Finally, wherever possible, sites are extended to the ends of their appropriate secondary structures and domain assignments are extended to their N- & C- termini.

STRUDL

STRUDL[28](STRUctural Domain Limits) unlike many of it's predecessor methods doesn't take into account any information on secondary structures and handles any number of domains made up of contiguous or non-contiguous chain segments. The core of the algorithm works on partitioning the 3-D structure of a protein into sets of residues such that the interactions between the sets is minimum. This is a graph partition problem which is known to be NP-hard, but the type of graphs representing residue interactions can be analyzed by graph heuristic, which gives an approximate solution. The authors have used Kernighan-Lin graph heuristic algorithm[29] to partition the protein structure into sets or residues with minimum interaction between them. The contact area between atoms is defined as the area of intersection of the Van Der Waals sphere around each atom and the faces of its weighted Voronoi polyhedron[30]. Using the contact-area measure and the Kernighan-Lin heuristic, the procedure identifies the partition with minimum contact area. The partition is then accepted or rejected based on various additional criteria pertaining to the expected properties displayed by a structural domain like size and compactness. When a partition is accepted, the procedure is repeated recursively to generate sub-structures until no further splits are viable.

PDP

PDP(Protein Domain Parser)[31] attempts to identify domain by hierarchical decomposition of a protein into smaller fragments based on the idea that inter-domain contacts are much more than intra-domain contacts. The program starts off by considering the whole protein chain as one domain consisting of one continuous fragment. At each step, PDP cuts a domain into two domains by two ways: (1) By a single cut in all possible sites in the polypeptide chain; or (2) by a double cut in spatially close (distance between Cα-atoms is <8Å), but are sequentially distant (more than 35 residues apart). After each attempt the number of contacts *nc*(i,j) between two new formed domains is counted and normalized by the size of the domains, and referred to as *nnc* (normalized number of contacts) as calculated in formula 1.4.

$$nnc(i,j) = nc(i,j)/(|i|^{\alpha}.|j|^{\alpha}) \tag{1.4}$$

Where |i| is the size of the domain i, $\alpha = 0.43$. The assumption on which this formula is based is that the expected number of contacts between two domains depends on their surface areas, which is proportional to $n^{2/3}$ for a spherical domain of n amino acids. If the minimum of the normalized contacts is less than the threshold, the cut is implemented and the recursive step is repeated for the two new domains. The threshold is computed specifically for each given domain and is equal to one half of the average contact density for the whole domain. After all cuts are made, the contacts between all domains are checked again and domains with large number of contacts (the normalized number of contacts is greater than two) are combined into one larger domain. At the last step PDP filters out all tiny domains (less than 30 amino acids).

DDomain

DDomain[32] divides a structure into domains using a normalized contact-based domain-domain interaction profile. The working assumption is that each structural domain corresponds to a continuous segment of its amino acid sequence, and the interaction between the domains is the weakest under a correct domain assignment. Domain-domain interaction is estimated by counting the number of contacts between the domains, where a contact is defined by the distance between two residue side-chain centers of mass within a distance cutoff of 6.5 Å. In order to facilitate comparison among domains, the interaction energy is normalized by the size of the individual domains. Thus obtained energy is termed as the interaction profile between the two domain candidates, with one domain being defined from residue 1 to i and another from i+1 to N$_r$, where N$_r$ is the total number of residues for the given protein. The lowest interaction profile E$^{Profile}(I_{min})$ is selected as it implies the weakest interaction between the two domain and is subjected to the following criteria, where I$_{min}$ is the location for two separated domains:

$$40 < I_{min} < N_r - 40 \tag{1.5}$$

$$E^{Profile}(I_{min}) \leq E^{low}_{cutoff} \tag{1.6}$$

$$E^{Profile}(j) - E^{Profile}(I_{min}) \geq E^{excess}_{cutoff} \tag{1.7}$$

for a continuous segment of length $> L_{cut}$ in both proposed domains (1:I_{min} and I_{min}+1:N_r)

Here, E^{low}_{cutoff} is the maximum allowed profile energy for a residue designated to be domain boundary, E^{excess}_{cutoff} is the minimum profile energy that is above the profile energy at the domain boundary, and L_{cut} is the minimum length of a continuous segment that satisfies the above condition. Finally, if two domains are found, the algorithm is repeated on each of the domain to see if further splits are possible. The above algorithm assumes that a domain is at least 40 residues long. Further, the three parameters E^{low}_{cutoff}, E^{excess}_{cutoff} & L_{cut} are determined by obtaining data from various data sets like CATH & SCOP and dividing it into two, training and testing data sets. For a given data set, the three parameters are obtained by optimizing the agreement between number of domains predicted and annotated in a given training set. Optimization is performed by simple grid search in step size of 0.01 for E^{low}_{cutoff}, E^{excess}_{cutoff} & 1 for L_{cut}.

DomainParser

DomainParser[33] uses a top-down graph theoretical approach for domain decomposition with a rigorous post processing step. In the first step, the protein structure is modelled as a graph by considering residues as nodes and the edges between them is drawn based on their proximity. The edge capacity(the maximum flow of an edge) is assigned keeping in mind that inter-domain interactions should be less than intra-domain or in other words the network is to be divided into two portions such that the edge capacity across the portions is minimum. Also, further rules like each domain must not have many discontinuous segments, splitting of β-sheet should be avoided while decomposing a protein and a β-strand should not be cut are also incorporated. Thus, edge capacity turns out to be a function of (a) the number of atom-atom contacts between residues; (b) the number of backbone-backbone contacts; (c) interactions across a β-sheet and within a β-strand. These parameters are optimized during the training stage of the algorithm. Then, the Ford-Fulkerson algorithm[34] for finding minimum cut/maximum flow is applied to bi-partition the graph. Briefly, the algorithm works by artificially adding a source and a sink node to the graph. A minimum cut is then calculated by finding the maximum flow from source to sink. A set of critical edges in the graph are identified by gradually increasing the flow of all edges and the ones with the least capacity are the ones which form the bottleneck. The removal of these edges stops the flow from source to sink. Thus, the nodes connected to the source form one of the possible domain, while the others connected to sink forms the other domain. This process is repeated a number of times by attaching the source and the sink to different nodes of the network. A set of minimum cuts is obtained and the algorithm is repeated recursively on the partitions obtained till the size of any one of the obtained domain falls below 80 or any of the obtained cuts stop meeting the criteria of the definition of a domain. The post-processing steps are used to evaluated the minimum cuts obtained and to refine the ones which are accepted. Some of the characteristic properties of domains which are used to evaluate the obtained partitions are (a) a domain should have at least 40 residues; (b) β-sheets are intact; (c) a domain should be compact; (d) the interface between two domains should be small; (e) the number of segments in a domain must be small.

A Hybrid Method for identification of structural domains(2014)

Hua et al[35]. used a hybrid method by combining two algorithms: Support Vector Machine(SVM) and k-means to identify the structural domains. Their algorithm takes into account, the density, length and dispersion of a domain in order to train SVM. The method works in two steps, the first step is where single and multi domain chains are classified by splitting a chain into two by using k-means and then computing the above mentioned properties of the two clusters and using them to train the SVM and then classify them. Once all the single-domain chains are filtered out, in the second step similar process is repeated to classify two-domain proteins from the rest of the multi-domain proteins. Furthermore, three and four domain proteins are identified based on the modularity of a cluster. The modularity contains intra-cluster and inter-cluster cohesiveness. Optimizing modularity by the following formulas can identify the number of domains.

$$Dis_{Intra-cluster} = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} d(c_k, n_i) \tag{1.8}$$

$$Dis_{Inter-cluser} = \sum_{k=1}^{K} d(c_k, center) \tag{1.9}$$

$$K = argmin(Dis_{Intra-cluster} + Dis_{Inter-cluser}) \tag{1.10}$$

The c_k, n_i, center are the kth-cluster's center, i^{th} node, the center of the whole chain respectively. The d(x,y) represents the Euclidean distance between x and y.

1.3.3 Manual & semi-automatic methods

Pfam

Pfam[36] is a database of large collection of protein families which contains evolutionary related sequences. Since domains are sequences which are conserved across evolution, the above mentioned families contains sequences which are similar in the context of domains. In essence, it captures the diversity of a set of evolutionary related domains. For each Pfam entry, a representative subset of the entire set of matching sequences are aligned to make a seed alignment. This seed alignment is used to construct a *profile* Hidden Markov Model(HMM) using the HMMER[37] software. The *profile* HMM is searched against a sequence database with all sequences scoring greater than or equal to a certain threshold[38] being classified as true members. These members are aligned to the *profile* HMM to generate the full alignment. Pfam entries which are being termed as related are grouped together into groups called **clans**. The relationship being measured by various means like HMM-HMM comparison, HMMER cross matches etc.

CATH

CATH is a semi-automatic method to classify protein domains in a hierarchical manner. The four main levels of its classification are class(C), architecture(A), topology(T) and homology(H). Class is the simplest level and defines the secondary structure composition of the protein. Topology captures the sequential arrangement, while architecture reflects the shape and the orientation of the protein. Homology puts together those domains which share a common topology and a similar functionality. Briefly, the database is created by the following steps. First, only well resolved crystal structures(3.0 Åresolution or better) and NMR structures from the Protein Data Bank(PDB) are used. Next, pairwise comparisons between the sequences of all the proteins selected for CATH are performed using Needleman & Wunsch algorithm. This is done as nearly three-quarters of the structures have identical sequences and are thus clubbed together. Completely identical proteins(100% sequence similarity and 100% overlap of structures) are grouped together into identical(I) families. Near-identical(N) families are subsequently created(>95% sequence similarity, at least 85% of larger protein equivalent to smaller). The S level grouping is created by clustering proteins having 35% or more sequence identity. The best resolved crystal structure of each family is used as the representative of that family. In the next step, domain boundaries for multi-domain proteins are assigned by using a consensus based approach between DOMAK[39], DETECTIVE & PUU[40] with the threshold set as having at least 85% overlap. Once the domains are identified, Class, Homology and Topology are assigned to each of the families in the I, N & S class using automatic procedures[41][42]. Finally, Architecture is assigned by manual inspection.

SCOP

The Structural Classification of Proteins(SCOP) is a manually created database which consider domains as the classifying unit. It classifies in a hierarchical manner and have four levels of classification. (a) Family: All proteins which either have a sequence similarity greater than or equal to 30% or those which have lower sequence similarities but are functionally similar are classified into the same family; (b) Superfamily: Those proteins which have low sequence similarity but their structure or function suggest a common evolutionary origin are placed under the same superfamily; (c) Common fold: Two or more proteins are said to be having a common fold if they have the same major secondary structure arrangement and also share similar topological connections; (d) Class: Based on the type of secondary structure a protein has, it can belong to one of the 5 structural classes: all-α, all-β, α/β, α+β & multi-domain proteins who have domains of different fold and for which no known homologues are present.

Chapter 2

Computational Methods

2.1 Introduction

Clustering or cluster analysis is a way by which entities which have similar properties are put together such that those belonging to the same group(cluster) are more similar to each other as compared to others in a different group. The properties by which these entities are grouped together can vary with the type of data that is under consideration. There is no single algorithm to identify clusters as the notion of what constitutes a cluster and how to identify them varies. Some of the commonly used notions defining a cluster are high density of entities within a cluster, small distances within group members and relatively large distances between clusters and so on. Thus, a wide variety of algorithms exist for clustering based on one or a combination of various definitions of a cluster. Cluster analysis provides a macro-level view of the data-set under observation. The primary task of this analysis is to group similar entities together and in the process work out the number of such groups. It has a lot of application in today's real world problems and is used in machine learning, pattern recognition, image analysis and bioinformatics to name a few. For example, shopping trends of users on e-commerce platforms can be used to compile them into groups and this data can be used to display specific set of advertisements to these obtained groups of shoppers.

Clustering finds a lot of its application in bioinformatics. For example, clustering can be used to study protein-protein interaction networks[43] where each node of the graph denotes a protein and the interaction between these nodes is captured by the edges. Clustering on this graph can be used to extract out the relevant modules for analysing protein-protein interactions. Another noteworthy example would be clustering of highly homologous sequences to be represented by a single sequence to reduce the size of the the large protein databases[44]. This potentially increases the speed of comparison and analysis amongst different such groups of protein sequences as the complexity is reduced. Clustering to understand gene expression[45], predicting protein structure[46] are some of its other use-cases in bioinformatics.

2.2 Clustering techniques

Since the definition of what a cluster is varies, different algorithms have emerged to achieve the goal of dividing a given data-set into meaningful clusters. These algorithms employ different cluster models to obtain clusters. A cluster model is basically the notion according to which a cluster is defined. It can be based on distance connectivity, density, statistical distribution etc. The following sections touch base upon various clustering models and their corresponding algorithms which are in popular use.

2.2.1 Connectivity models

These models are based on the notion that points closer to each other exhibit more similarity than those which are further apart. The **Hierarchical clustering**[47] algorithm is based on this model and employs two techniques to cluster. One of the approach is the *top-down* or *divisive* one which considers the entire data-set as a single cluster and keeps on recursively dividing it into smaller clusters based on connectivity in a hierarchical manner such that any obtained cluster is a sub-set of the parent cluster from which it was extracted in the previous step. This process is continued till a predefined criteria is met which could be anything like the number of clusters, minimum cluster size, threshold connectivity among clusters or a combination of one or more of the above. The other approach is is *bottom-up* or *agglomerative* in which every data-point in the given data-set is considered as a cluster of its own. At each step, these clusters are merged based on connectivity and this is recursively done till the merging threshold or criteria is met. In general, the merges and splits are determined in a greedy manner and the results of hierarchical clustering is represented by a dendrogram as show in figure 2.1.

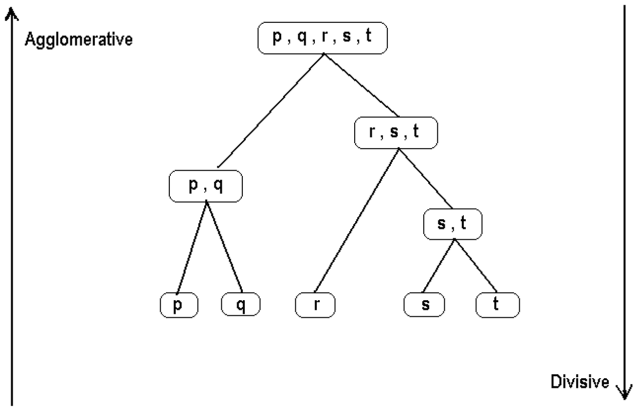

Figure 2.1: Agglomerative and divisive hierarchical clustering

2.2.2 Centroid models

In centroid models, clusters are represented by central vector which may not be a part of the data-set. Simply put, the centroid of a cluster is used to represent the entire cluster and is used to calculate similarity with other points in the data-set. One of the downsides of centroid models is that most of the algorithms based on it require the number of clusters into which the data is to be partitioned beforehand. The k-means clustering algorithm is based on the centroid model and is discussed in detail in section 2.4.

2.2.3 Distribution models

Distribution models are very closely related to statistics. Objects belonging to the same cluster are the ones which have a similar probabilistic distribution. Gaussian mixture models are one of the well known distribution models used for clustering. Here, the data set is usually modelled by a fixed number of Gaussian distribution that are initialized randomly and parameters are iteratively optimized to fit better to the data set. Distribution based models suffer from the problem of over-fitting, unless constraints are put on the model. Also, these models makes a strong assumption that the give data will fit in an already existing mathematical model, for example Gaussian distribution and this often leads to poor results.

2.2.4 Density models

2.2.5 Graph based models

2.3 Identification of number of domains by using physical properties of a protein & SVM

2.4 K-means clustering algorithm for predicting domain boundaries

2.5 Implementation details

2.5.1 Construction of data set

2.5.2 Libraries & tools

Chapter 3

Results & Discussion

Chapter 4

Conclusion

Bibliography

[1] Bork, P. Shuffled domains in extracellular proteins. *FEBS Letters* **1991**, *286*, 47 –54.

[2] Murzin, A. G.; Brenner, S. E.; Hubbard, T.; Chothia, C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol.* **1995**, *247*, 536–540.

[3] Orengo, C. A.; Martin, A. M.; Hutchinson, G.; Jones, S.; Jones, D. T.; Michie, A. D.; Swindells, M. B.; Thornton, J. M. Classifying a Protein in the CATH Database of Domain Structures. *Acta Crystallographica Section D* **1998**, *54*, 1155–1167.

[4] Wetlaufer, D. B. Nucleation, Rapid Folding, and Globular Intrachain Regions in Proteins. *Proceedings of the National Academy of Sciences* **1973**, *70*, 697–701.

[5] Veretnik S, W. S., Gu J Identifying Structural Domains in Proteins. *Genny Gu and Philip Bourne Structural Bioinformatics. Second edition. Wiley-Blackwell* **2009**, 485–513.

[6] Kikuchi, T.; Némethy, G.; Scheraga, H. A. Prediction of the location of structural domains in globular proteins. *Journal of Protein Chemistry* **1988**, *7*, 427–471.

[7] Wheelan, S. J.; Marchler-Bauer, A.; Bryant, S. H. Domain size distributions can predict domain boundaries. *Bioinformatics* **2000**, *16*, 613–618.

[8] Doolittle, R. F. The Multiplicity of Domains in Proteins. *Annual Review of Biochemistry* **1995**, *64*, 287–314, PMID: 7574483.

[9] Marchler-Bauer, A.; Anderson, J. B.; Cherukuri, P. F.; DeWeese-Scott, C.; Geer, L. Y.; Gwadz, M.; He, S.; Hurwitz, D. I.; Jackson, J. D.; Ke, Z. et al. CDD: a Conserved Domain Database for protein classification. *Nucleic Acids Research* **2005**, *33*, D192196.

[10] Liu, J.; Rost, B. Sequence-based prediction of protein domains. *Nucleic Acids Research* **2004**, *32*, 3522–3530.

[11] Altschul, S. F.; Madden, T. L.; Schäffer, A. A.; Zhang, J.; Zhang, Z.; Miller, W.; Lipman, D. J. Gapped BLAST and PSI-BLAST: a new

generation of protein database search programs. *Nucleic Acids Research* **1997**, *25*, 3389–3402.

[12] Bairoch, A.; Apweiler, R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Research* **2000**, *28*, 45–48.

[13] Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Research* **2000**, *28*, 235–242.

[14] Werbos, P. J. *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*; Wiley-Interscience: New York, NY, USA, 1994.

[15] Sander, C.; Schneider, R. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Structure, Function, and Bioinformatics* **1991**, *9*, 56–68.

[16] Vihinen, M.; Torkkila, E.; Riikonen, P. Accuracy of protein flexibility predictions. *Proteins: Structure, Function, and Bioinformatics* **1994**, *19*, 141–149.

[17] Suyama, M.; Ohara, O. DomCut: prediction of inter-domain linker regions in amino acid sequences. *Bioinformatics* **2003**, *19*, 673–674.

[18] Bondugula, R.; Lee, M. S.; Wallqvist, A. FIEFDom: a transparent domain boundary recognition system using a fuzzy mean operator. *Nucleic Acids Research* **2009**, *37*, 452–462.

[19] Keller, J. M.; Gray, M. R.; Givens, J. A. A fuzzy K-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics* **1985**, *SMC-15*, 580–585.

[20] Portugaly, E.; Harel, A.; Linial, N.; Linial, M. EVEREST: automatic identification and classification of protein domains in all protein sequences. *BMC Bioinformatics* **2006**, *7*, 277.

[21] Smith, T.; Waterman, M. Identification of common molecular subsequences. *Journal of Molecular Biology* **1981**, *147*, 195 –197.

[22] DC, P. Past and Present. *British Biochemistry. London Academic Press* **1970**, 11–28.

[23] Ooi T, N. K. Conformation of Biological Molecules and Polymers. *New York: Academic Press* **1973**, 173–187.

[24] Rossman MG, L. A. Letter: recognition of structural domains in globular proteins. *J Mol Biol* **1974**, *85*, 177–181.

[25] Swindells, M. B. A procedure for detecting structural domains in proteins. *Protein Science* **1995**, *4*, 103–112.

[26] Swindells, M. B. A procedure for the automatic determination of hydrophobic cores in protein structures. *Protein Science* **1995**, *4*, 93–102.

[27] Chothia, C. The nature of the accessible and buried surfaces in proteins. *Journal of Molecular Biology* **1976**, *105*, 1 –12.

[28] Wernisch L., W. S., Hunting M. Identification of structural domains in proteins by a graph heuristic. *Proteins* **1999**, *35*, 338–352.

[29] Kernighan, B. W.; Lin, S. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal* **1970**, *49*, 291–307.

[30] Edelsbrunner, H. *Algorithms in Combinatorial Geometry*; Springer-Verlag New York, Inc.: New York, NY, USA, 1987.

[31] Alexandrov, N.; Shindyalov, I. PDP: protein domain parser. *Bioinformatics* **2003**, *19*, 429–430.

[32] Zhou, H.; Xue, B.; Zhou, Y. DDOMAIN: Dividing structures into domains using a normalized domain–domain interaction profile. *Protein Science* **2007**, *16*, 947–955.

[33] Xu, Y.; Xu, D.; Gabow, H. N. Protein domain decomposition using a graph-theoretic approach. *Bioinformatics* **2000**, *16*, 1091–1104.

[34] Ford, D. R., L. R.; Fulkerson Maximal flow through a network. *Canadian Journal of Mathematics* **1956**, *8*, 399–404.

[35] Hua, Y.; Zhu, M.; Wang, Y.; Xie, Z.; Li, M. A hybrid method for identification of structural domains. *Scientific Reports* **2014**, *4*, 7476 EP –, Article.

[36] Finn, R. D.; Bateman, A.; Clements, J.; Coggill, P.; Eberhardt, R. Y.; Eddy, S. R.; Heger, A.; Hetherington, K.; Holm, L.; Mistry, J. et al. Pfam: the protein families database. *Nucleic Acids Research* **2014**, *42*, D222230.

[37] Finn, R. D.; Clements, J.; Eddy, S. R. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research* **2011**, *39*, W2937.

[38] Punta, M.; Coggill, P. C.; Eberhardt, R. Y.; Mistry, J.; Tate, J.; Boursnell, C.; Pang, N.; Forslund, K.; Ceric, G.; Clements, J. et al. The Pfam protein families database. *Nucleic Acids Research* **2012**, *40*, D290301.

[39] Siddiqui, A. S.; Barton, G. J. Continuous and discontinuous domains: An algorithm for the automatic generation of reliable protein domain definitions. *Protein Science* **1995**, *4*, 872–884.

[40] Holm, L.; Sander, C. Parser for protein folding units. *Proteins: Structure, Function, and Bioinformatics* **1994**, *19*, 256–268.

[41] Michie, A. D.; Orengo, C. A.; Thornton, J. M. Analysis of Domain Structural Class Using an Automated Class Assignment Protocol. *Journal of Molecular Biology* **1996**, *262*, 168 –185.

[42] Orengo, C.; Brown, N.; Taylor, W. Fast structure alignment for protein databank searching. *Proteins* **1992**, *14*, 139—167.

[43] Brohée, S.; van Helden, J. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics* **2006**, *7*, 488.

[44] Li, W.; Jaroszewski, L.; Godzik, A. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics* **2001**, *17*, 282.

[45] Datta, S.; Datta, S. Evaluation of clustering algorithms for gene expression data. *BMC Bioinformatics* **2006**, *7*, S17.

[46] Gront, D.; Kolinski, A. HCPM—program for hierarchical clustering of protein models. *Bioinformatics* **2005**, *21*, 3179.

[47] Johnson, S. C. Hierarchical clustering schemes. *Psychometrika* **1967**, *32*, 241–254.