2016

PREDICTION OF CALMODULIN-BINDING PROTEINS USING SHORT LINEAR MOTIFS

Mrinalini Pandit
University of Windsor

Prediction of Calmodulin-Binding Proteins using Short Linear Motifs

by

Mrinalini Pandit

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2016

Prediction of Calmodulin-Binding Proteins using Short Linear Motifs

by

Mrinalini Pandit

Approved By:

Dr. Huapeng Wu
Electrical and Computer Engineering

Dr. Alioune Ngom
Computer Science

Dr. Luis Rueda, Advisor
Computer Science

May 13, 2016

Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Calmodulin-binding (CAM-binding) proteins are one of the most recently studied cases of protein interactions. Calmodulin is a calcium-binding protein that is a major transducer of calcium signal that forms an intricate network of loops to control the location, amount and effect of calcium influx. CAM-binding proteins are a varied group of targets and their interactions with Calmodulin can be subdivided into calcium-independent and calcium dependent ways of binding and regulation.

Prediction of CAM-binding proteins plays a very important role in the fields of biology and biochemistry, because Calmodulin binds and regulates a multitude of various protein targets affecting different cellular processes such as inflammation, metabolism, muscle contractions, intracellular movement, short-term and long-term memory, nerve growth and immune response.

Prediction of CAM-binding proteins involves intensive analysis of the protein sequences and their binding sites. In this thesis, I propose a model for prediction of CAM-binding proteins using Short Linear Motifs (SLiMs). The model uses information in these short stretches of the protein sequences as various features to predict and analyze Calmodulin proteins as calcium-independent and calcium-dependent. Prediction of Calmodulin proteins using SLiM profiles has potential for help in the modulation and regulation of cellular processes such as proliferation and apoptosis. Prediction and analysis of CAM-binding proteins can also help understand and broaden the knowledge of signaling networks.

Dedication

I dedicate this thesis to my family; my partner, my parents and my elder sister for their constant and endless love, prayers, support and motivation. Without their wishes, inspiration and hard work I would not be able to complete my Master's degree and research.

Acknowledgements

I would like to express my sincere appreciation and gratitude to my supervisor Dr. Luis Rueda for his continuous encouragement, support, guidance and interesting discussions throughout the duration of study. Without his assistance, I would never have been able to present my research.

I would also like to thank Dr. Mina Maleki for her consistent support and guiding me throughout in this research field and providing me with immense knowledge that helped me broaden and learn new concepts in the field of computational biology.

I would like to thank Dr. Nick Carruthers, Dr. Paul Stemmer from the Wayne State University for their help in providing the dataset and guiding on CaM and non CaM-binding proteins. I would also like to express my thankfulness to the team of MEME for their prompt response in resolving queries related to MEME and SLiMs.

I would also like to thank Dr. Alioune Ngom for his valuable inputs in my thesis which helped in exploring more deeper areas of Calmodulin.

I would also like to thank Dr. Huapeng Wu (External reader), Department of Electrical and Computer Engineering and Dr. Scott Goodwin (Defense chair), School of Computer Science for being the committee members and providing their precious time and efforts.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Bioinformatics is a vital field out of various areas in biology and biochemistry. In computational molecular biology, bioinformatics techniques, for example, image and signal processing permit extraction of valuable results from large raw data. In the area of genetics and genomics, it helps in sequencing and annotating genomes and their known transformations. It plays an imperative role in the concept of data mining of biological literature and the development of biological and gene ontologies to arrange and inquire biological information. It likewise assumes a critical role in the examination of quality and protein sequences and regulation. At a more integrative level, it examines and lists the biological pathways and systems that are a critical part of systems biology. In structural biology, it helps in the re-enactment and display and formation of DNA, RNA, and protein structures and sub-atomic or molecular interactions [19].

The study of bioinformatics involves the analysis and interpretation of various types of data such as nucleotide and amino acid sequences, protein domains, and protein structures. It also includes the development of new algorithms and statistical data that provide information about relationships among members of large datasets. For example, techniques to locate a gene within a sequence, to predict protein structure and/or its function, and to cluster or group protein sequences into families of related sequences.

Protein structure prediction is an important problrm of bioinformatics. The amino acid sequence of a protein, called the primary structure, can be easily determined from the sequence on the gene that codes for it. In broad scenarios, this primary structure uniquely determines a structure in its native environment. Knowledge of this

structure is vital in understanding the function of the protein. Structural information is usually classified as one of secondary, tertiary and quaternary structure. However, a viable general solution to predictions of protein structures and their interactions remains an open issue. Many efforts have so far been directed towards heuristics approaches that have been successful in most of the cases.

1.1 Proteins

Proteins are polypeptides, which are made up of many amino acids inter-linked together as a linear chain. The structure of an amino acid contains an amino group, a carboxyl group, and an R group which is usually carbon based and gives the amino acid its specific features and properties. These properties determine the interactions between atoms and molecules within and in between the proteins and other structures[1].

Proteins form the basis of life. They regulate a variety of activities in all known organisms, from replication of the genetic code to transportation of oxygen, and are generally responsible for regulating the cellular structures and systems and determining the phenotype of an organism. Proteins accomplish their tasks in the body by three-dimensional tertiary and quaternary interactions between various substrates. The functional properties depend upon the protein's three-dimensional structure. These 3D structures arise because particular sequences of amino acids in a polypeptide chain fold to generate, from the linear chains, compact domains within specific structures. The folded domains either serve as modules for larger assemblies or they provide specific catalytic or binding sites.

Proteins consist of an extremely heterogeneous class of biological macromolecules. Many are unstable when they are not located in their native environments, i.e., various cell compartments and extracellular fluids. The native (folded) proteins are only marginally stable under physiological conditions. Other forces such as hydrophobic effects, electrostatic interactions, and hydrogen bonding act more as stabilizing factors and are the main factors in driving the protein folding process.

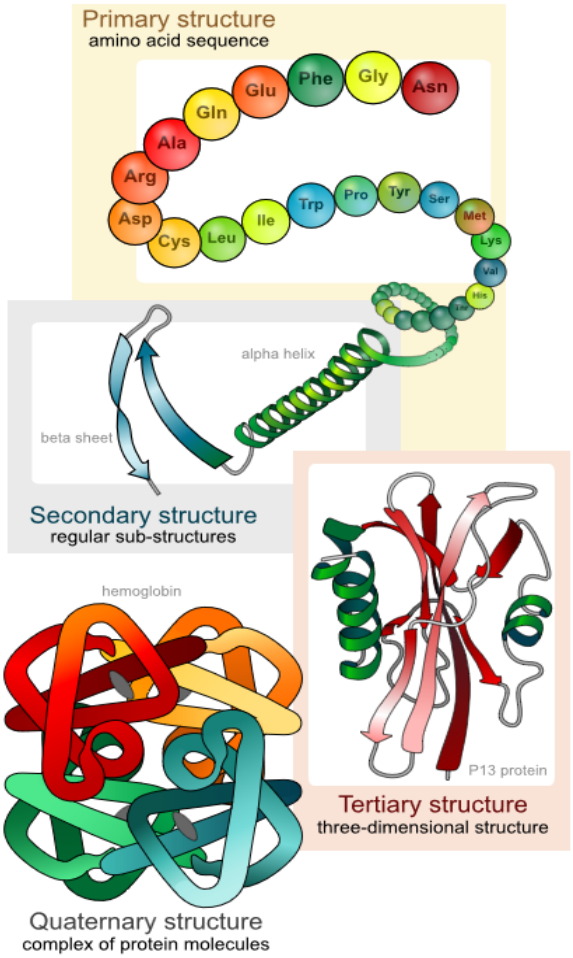

Figure 1.1: Protein Structure, from primary to quaternary (Figure extracted from [1]).

Figure 1.1 shows the discrete level of the protein structure. The first level is the amino acid sequence of the chains. The second level includes the helical segments of the protein as well as the connecting loops. The third level includes the complete three-dimensional organization of each of the chains. Finally, the fourth level includes the arrangement of the different chains via interactions.

1.2 Protein-Protein Interaction

The subject of protein-protein interactions represents an immense ensemble of results from biological, biochemical and biophysical studies carried out to date. Protein-protein interactions are functional at almost every level of the cell function, in the structure of sub-cellular organelles, the transport systems across the various biological membranes, muscle contraction, and signal transduction, regulation of gene expression. Protein-protein interactions have been entailed in a number of neurological disorders [1].

Because of their importance in development and disease identification and curation, prediction of protein interactions has been the object of intense research for many years. It has emerged from these studies that nature has employed many instances of strategies of inter-mixing and matching of domains that specify particular classes of protein-protein interactions, modifying the amino acid sequence in order to confer specificity for particular target proteins. The regulation of cell function brought about by the interactions of these proteins is balanced by the relative relationship of the various protein partners and the modulation of these affinities such as by the binding of ligands, other proteins, nucleic acids, ions such as Ca^{2+}, and covalent modification, such as specific phosphorlyation or acetylation reactions. Specificity and the strength of signal transduction is encoded by the exact amino acid sequence of the domain, and it is this relationship between sequence, structure, dynamics, energetics and function that constitutes the fundamental issue for the principles of protein-protein interactions [20].

1.3 Motifs

Motifs are short regions of proteins and frequently are the most conserved regions of the domains. They are patterns that describe a short, contiguous stretch of proteins sequences. They are either sequential or structural in character.

1.3.1 Sequence Motifs

A sequence motif is a nucleotide or a sequence pattern of amino-acid that is widely distributed and is speculated to have some biological significance. It is the motif formed by the arrangement of amino acids in three-dimensional structure [2].

Some sequence motifs are short, recurring patterns in proteins that are assumed to have some biological function. They suggest sequence-specific binding sites for proteins such as nuclease and transcription factors (TF). They have active involvement in important processes at the RNA level, such as ribosome binding, mRNA processing (i.e., splicing, editing, polyadenylation) and transcription termination [3] and now for the prediction of Calmodulin-binding proteins.

1.3.2 Structural Motifs

A structural motif is a short segment of protein structure with dissimilar functions. They are conserved in large number of different proteins. Their role is either structural or functional in nature within the protein sequence.

Structural motifs are supersecondary structures with combinations of alpha-helices and beta-structures inter-connected through loops. These structures are simple as alpha-alpha (i.e., two alpha helices that is connected by a loop) or Beta-Beta (which is two beta-strands interlinked by a loop) and Beta-Alpha-Beta (Beta-strand connected to an alpha helix that is also linked to other beta strands, by loops) or more complexes structures [5].

1.3.3 Short Linear Motifs (SLiMs)

Short Linear Motifs (SLiMs) are short stretches of protein sequence that arbitrate protein protein interactions [1][2]. Protein sequences contain short, conserved motifs that are involved in arbitration of binding sites. These motifs are linear, hence the three-dimensional formation is not required in the protein structure. The conservation of these motifs varies as some may be highly conserved while others may allow substitutions that retain only a certain pattern change (such as mutations, deletions and insertions) within the protein sequences across the motif. SLiMs are short sequences of 3 to 20 continuous amino acids.

1.4 Calcium

Calcium is the most common mineral element found in the human body. Our body uses only a substantial amount of calcium and these are in form of calcium ions [6]. Calcium has its own functioning activities out of which it is actively involved in signaling process of cells and other cellular activities, control of muscle contractions, nervous system and even in the reproduction system of organisms. The calcium levels in the cells are kept at lower level (1000 - 10,000) than stored in the blood. So when the calcium is released from the blood to perform various activities in the cell, they start to interact with proteins that sense calcium. This action triggers biological changes in the human body. These calcium sensing proteins are called Calmodulin. They help in regular biological activities within the organism for their smooth functioning [6].

Calcium ions play an important role in the metabolism and physiology of eukaryotes. External signals, such as hormones, light, stress or origination or development of any diseases in the body, can often lead to transient increases in calcium concentrations within the cell [7]. These increased calcium concentrations lead to calcium binding by regulatory proteins, which turn the calcium signals into a biological response. There are many such regulatory proteins that bind calcium, which together form an complex network of feedback loops to control the location, amount and effect of calcium inflow. Calmodulin is one such calcium-binding protein that is considered a major transducer of calcium signals [6].

1.5 Calmodulin (CaM)

Calmodulin (an abbreviation for **Cal**cium-**Modul**ated prote**in**) is a multifunctional calcium-binding messenger protein [2]. Calmodulin (CaM) is omnipresent, a calcium-binding protein that binds to and regulates a large number of different protein targets. These functional changes affect the various different cellular functions and processes [4].

Calmodulin intercedes processes such as inflammation, metabolism, apoptosis, muscle contraction, intracellular movement, short-term and long-term memory, nerve growth and the immune response. CaM is present in many cell types and can have different subcellular locations which includes the cytoplasm, inside of the cell, or associated with the plasma or organelle membranes. Many of the proteins that CaM

binds to are unable to bind calcium on their own. Thus they use CaM on a calcium sensor and signal transducer for the binding process. CaM undergoes a conformational change upon binding to calcium that further enables it to bind to specific proteins for a specific response or purpose. CaM can bind up to four calcium ions, and can undergo post-translational modifications, such as phosphorylation, acetylation, methylation, each of which can potentially modulate various Calcium actions in the organisms [8].

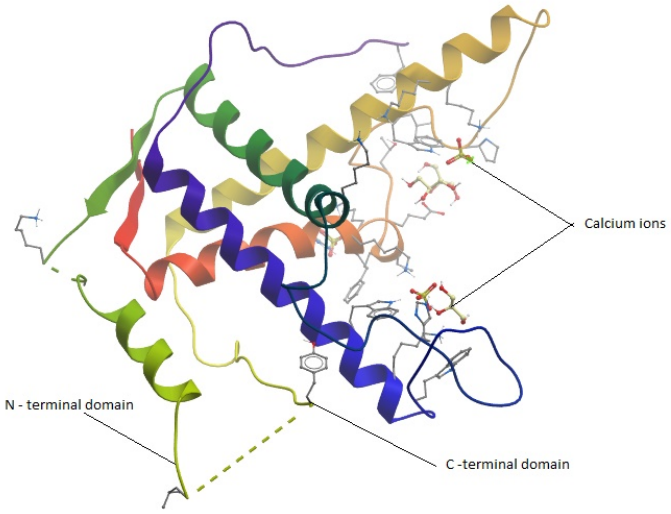

Figure 1.2: CaM-binding protein with C-terminal and N-terminal domain. This figure was created using the ICM browser [21] fom the PDB [22] structure of Cam-binding protein Q15413.

Figure 1.2 explains the process of the release of calcium channels that are located in the C-terminal region of the CaM-binding protein. It indicates that Ca^{2+} binding causes a larger conformational change and is used for cell signaling.

1.6 Machine Learning

Machine learning is a sub domain of computer science that has evolved from the study of pattern recognition and computational learning theory in artificial intelligence. Machine

learning explores the construction and study of algorithms that can learn from and make predictions on raw data. Such algorithms operate by building a model from sample or raw inputs in order to make data-driven predictions or decisions rather than following strictly following static program instructions.

Machine learning is engaged in a range of computing tasks where designing and programming explicit algorithms is infeasible and complicated. Examples of applications include spam filtering, optical character recognition (OCR), search engines and computer vision. Machine learning is sometimes combined with data mining. Machine learning and pattern recognition "are said to be the two evolving facets of the same field." [22]

1.7 Pattern Recognition

Pattern recognition on the other hand, is a branch of machine learning that focuses on the identification of patterns and regularities in data, although in some cases it is considered to be nearly synonymous with machine learning. Pattern recognition systems are in many cases trained from labeled "training" data (supervised learning), whereas when no labeled data are available other algorithms can be used to discover previously unknown patterns (unsupervised learning) [23].

In machine learning, pattern recognition is the assignment of a label to a given input value. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of classes (for example, determine whether a given email is "spam" or "non-spam"). However, pattern recognition is a more general problem that encompasses other types of outputs as well. Other examples are regression, which assigns a real-valued output to each input; sequence labeling of data, which assigns a class to each member of a sequence of values; and parsing, which assigns a parse tree to an input sequence, describing the syntactic structure of the sequence.

Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform similar matching of the inputs, taking into account their statistical variation.

In simple terms, pattern recognition involves the study of making machines observe the environment, make them learn to differentiate patterns of interest and finally make capable of taking reasonable decisions about the classification of these patterns just as human minds are capable of.

1.8 Feature Generation and Classification

A feature is defined as any distinctive aspect, quality or characteristic which, can be symbolic (i.e., color) or numeric (i.e., height). The combination of these features is represented as a *d*-dimensional column vector called a feature vector. The *d*-dimensional space defined by the feature vector is called feature space. Objects are represented as points in the feature space. This representation is called a scatter plot.

A pattern is defined as a composition of features that are characteristic of an individual. In classification, a pattern is a pair of variables (x,w) where x is a collection of observations or features (feature vector) and w is the concept behind the observation (label). The quality of the feature vector is related to its ability to discriminate samples from different classes observed in Figure 1.3. These samples from the same class have similar feature values and while samples from different classes having different feature values [23].

Figure 1.3 depicts the characterization of the features. When the feature selection method chosen extracts the features quite well, the good and bad features can be distinguished and further used for classification. The figure on the left has good features selected and hence the classification will generate good and better results as compared to the figure on the right since its features are clustered. This makes the classification process complex and the classification results may not be good or as expected.

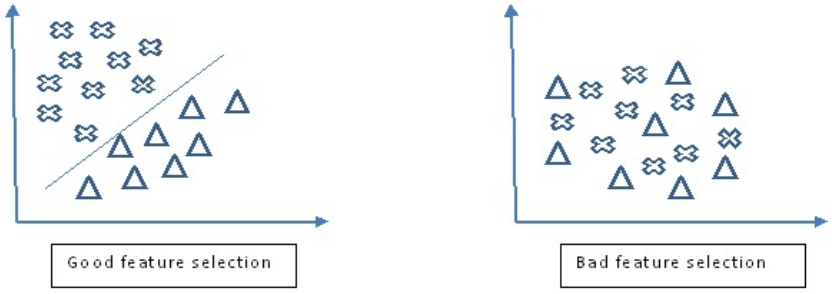

Figure 1.3: Characteristic (feature); the distinction between good and poor features.

The conceptual bonding between feature extraction and classification is complex: The feature extractor extracts information from the input data which is relevant for further classification.

The goal of the feature extractor is to perform a type of filtration in the patterns of objects. This filtration can be separating the similar qualities or features of objects from the dissimilar ones. This will help in good classification of the objects as the irrelevant features are removed keeping only the discriminative features that are unique.

The task of the classifier is to use the feature vector that results from the selection of features performed in the feature extractor stage and make decisions to classify in a particular category. In reality, accurate classification is not feasible and hence the classification methods generally calculate a probability value and based on this probability value determine the class which the object can be categorized to. This layer of abstraction provided by the feature vectors of the input data leads to the development of largely domain-independent theory of classification [23].

Practically speaking, it is not always possible to determine the best features for a particular input. For instance, the assumption made that the the value of the missing feature is zero or the average of the values for the patterns is considered to be non-optimal. Thus normally, the features are determined by generalizing them into a best fit for the classification of the object.

1.9 Motivation and Objectives

The large datasets of protein interactions provide a rich resource for the discovery of Short Linear Motifs (SLiMs) that recur in unrelated proteins. However, existing methods for estimating the probability of motif recurrence and identifying the Calmodulin-binding proteins are mostly inaccurate as motifs containing different numbers of non-wildcard positions are not comparable. Thus the objective is to develop more efficient methods and explore the potential predetermination of computationally efficient approximations methods. The discovery of new and unknown SLiMs is also a motivation to distinguish CaM-binding proteins from the whole group of proteins existing in human organisms.

Many biochemical and biological investigations have focused on determining CaM-binding proteins from the rest of the proteins. These concerted efforts have resulted in identification of many proteins that bind to Calmodulin, though

identification of CaM-binding proteins is still a sloppy and slow exploration and experimentation [8].

1.10 Problem Definition

The calcium-binding protein called as calmodulin (CAM) directly binds to membrane transport proteins for modulating their function in response to the changes in intracellular calcium concentrations. Calmodulin recognizes and binds to a large collection of target sequences, and hence makes it difficult to identify the CaM-binding proteins from the remaining targets. This identification of Calmodulin-binding proteins requires intensive sequence and extensive biochemical analysis.

Therefore, the main focus of this thesis is to identify CaM-binding proteins from other proteins present in eukaryotes using Short Linear Motifs (SLiMs) and further classify the resulting proteins as being either Calcium-binding or non Calcium-binding.

1.11 Thesis Organization

This thesis has a total of eight chapters. Chapter 2 provides details of Calmodulin-binding proteins, Calmodulin-binding sites and motifs. Chapter 3 explains the Short linear motifs (SLiMs).Chapter 4 discusses the various motifs identification methods and different SLiM identification tools. It also focuses on the usefulness and configuration of a SLiM identification tool - Multiple EM for Motif Elicitation (MEME). Chapter 5 focuses on different feature selection and extraction methods that are used for prediction. It also includes various classifiers used for the classification process. Chapter 6 describes the entire implementation of the proposed model, and Chapter 7 describes the results and discussions of the implementation. Finally, Chapter 8 depicts the conclusions of the thesis, its use in the field of computational biology along with the future work.

Chapter 2

Calmodulin-Binding Proteins

Calmodulin is a CALcium MODULated proteIN. It is abundant in the cytoplasm of all higher cells and is highly conserved through evolutions. Calmodulin acts as an intermediary protein that senses calcium levels and relays signals to various calcium-sensitive enzymes, ion channels and other proteins for smooth and efficient functioning. Calmodulin is a small "dumbbell-shaped" [2] protein that is composed of two globular domains which are connected together by a flexible linker connection. Each end binds to two calcium ions [2].

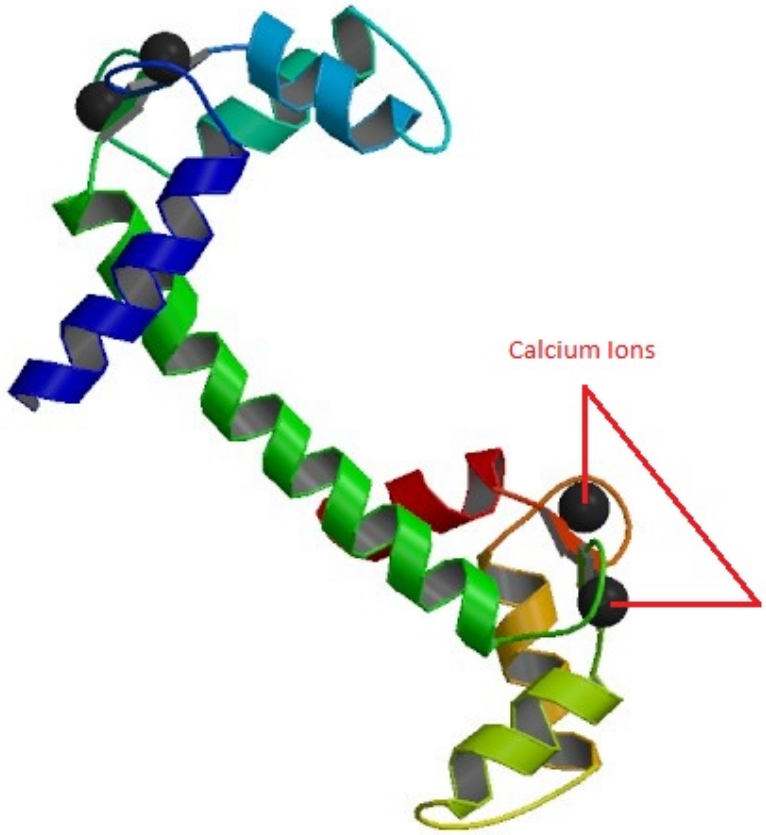

Figure 2.1: PDB entry 3cln has all four sites filled with calcium ions and the linker connection forms a long alpha helix separating the two calcium-binding domains.

Calmodulin is an intracellular calcium receptor present universally in eukaryotes. It is capable of regulating biological activities of many cellular proteins and trans-membrane ion transporters mainly in a Ca^{2+}-dependent order. The intracellular calcium level rises to 10^{-5} M, four Ca^{2+} ions bind to calmodulin, and this Ca^{2+}-Calmodulin complex binds the target proteins, initiating various signalling cascades [12].

Calmodulin has four EF-hand motifs that change conformation when binding to calcium ions. Each EF-hand motif contains two alpha helices connected by a 12-residue

loop. The calcium ion binds to the loop region and changes the relative positions of the alpha helices. In absence of calcium, the alpha-helices in the EF-hand motif of the Calmodulin are positioned almost parallel to each other. This is known as the closed configuration [12].

2.1 Calmodulin Structure

Calmodulin contains four similar high-affinity calcium binding sites, as shown in Figure 2.2 of PDB entry 1cll. The calcium ions are represented in the diagram in round yellow color. The calcium-binding motif consists of a loop with two alpha helices located at its sides.

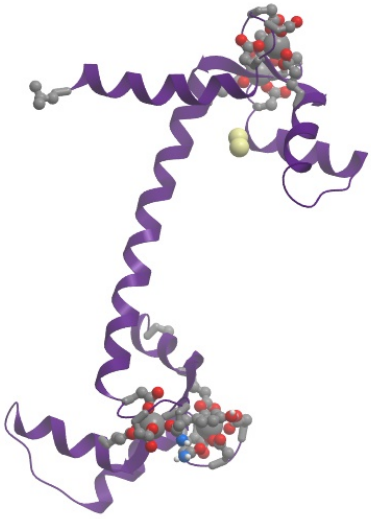

Figure 2.2: Structural diagram of PDB entry 1cll. The figure was created using ICM browser [21] using the PDB [24] file 1cll.

Figure 2.3 shows the positively-charged calcium ion that is encircled in a loop by negatively-charged side chains of three aspartates and one glutamate and one oxygen atom [15][6].

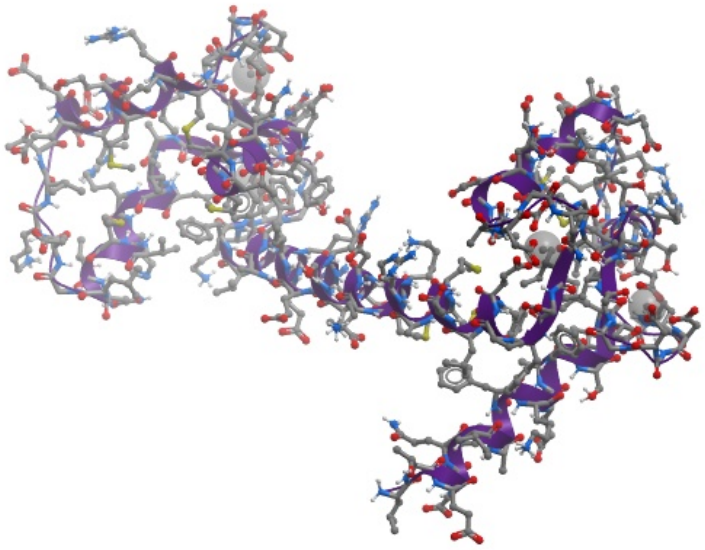

Figure 2.3: Backbone diagram of PDB entry 1cll. The figure was created using the ICM browser [21] with the PDB [24] file 1cll.

2.2 Versatility of Calmodulin

Calmodulin's target proteins have various shapes, sizes and sequences. They are involved in a wide variety of functions. For instance, calcium-bound calmodulin configures into a very important subunit for the regulatory enzymes, phosphorylase kinase. This formation helps in the breakdown of glycogen which is a form of glucose necessary for the energy breakdown in organisms. Calmodulin binds and triggers other kinases and phosphatases which help in cell signaling, transportation of ions between cells and cell expiration. The structure shown in Figure 2.4 is taken from the PDB entry 1cfd. It depicts calmodulin without calcium. The structure on the right is taken from PDB entry 1cll which shows the changes that occur after Calmodulin binds to calcium. The shape formed out of the binding is not pertaining to any chemical change and is quite generic. This makes Calmodulin a "skilled" regulatory protein and so its targets

need not require any particular sequence of amino acid nor specific binding motifs [6].

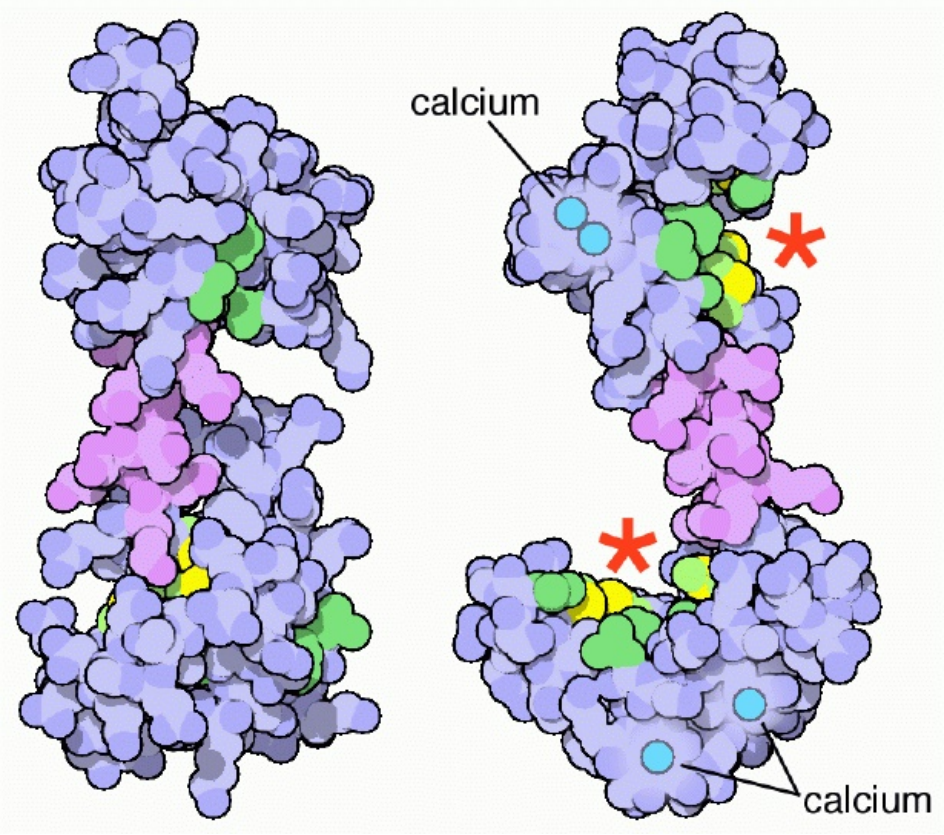

Figure 2.4: Protein motion in calmodulin: without calcium (image on the left) and with calcium (image on the right). The binding sites, which are the regions of binding with its target proteins, are marked with stars. This figure was downloaded from Wikipedia [2].

2.3 Calmodulin's Tractability

Calmodulin is very flexible in iself as its calcium binding domains are very tractable. However, Calmodulin becomes extremely flexibile when it binds to its target proteins and interacts with them wrapping the target proteins completely in itself. This can be

observed in Figure 2.5. It is shown that Calmodulin wraps around its target protein with its two domains surrounded on both sides. The two structures on the left represent Calmodulin binding to two different targets which are Calmodulin-dependent protein kinase II-alpha at the top (its PDB entry is 1cm1) and myosin light chain kinase at the bottom (its PDB entry is 2bbm). The structure on the right is the internal binding of Calmodulin with its target. This is shown in red.

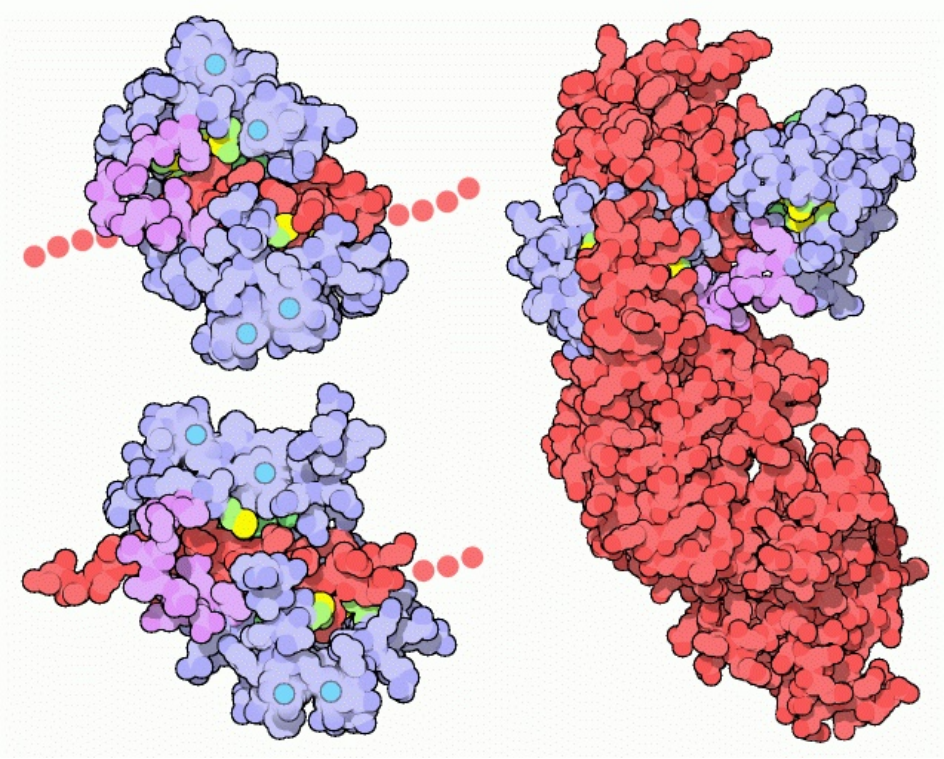

Figure 2.5: Complexes of Calmodulin with target proteins: with peptides from Calmodulin-dependent protein kinase II-alpha (extreme top left) and myosin light chain kinase (extreme bottom right), and with anthrax edema factor (right).

2.4 Functionality

The generic function of Calmodulin is to binds calcium ions which in turn binds a target protein thereby changing its functionality and activity. The binding of calcium is done by the 4 EF-hand domains. A basic EF hand consists of two perpendicular alpha helices and a 12-residue loop region between them. The residues 1, 3, 5, 7, 9 and 12 of the loop region handle for calcium binding. Each of these residues (with the exception of 12) contributes a sidechain or backbone oxygen atom which is needed for the binding of calcium. On the contrary, residue 12, which is usually a Glu or Asp, alters both oxygens from its side chain carboxylic acid group. In the calcium-bound EF-hand, calcium ions consequently bind to 7 oxygen atoms in a pentagonal bi-pyramidal shape. Normally, Calcium ions bind to a pair of EF-hands at a time. This means that Calmodulin binds to 0, 2, or 4 calcium ions simultaneously [15].

Calcium binding causes Calmodulin to develop a conformational change which allows Ca^{2+}-CaM to bind to its targets and thereby extract several responses. When Calmodulin binds to calcium, its hydrophobic regions break out in the domain on the surface. These hydrophobic surfaces interact with the hydrophobic regions of Calmodulin's target proteins. Calmodulin behaves as a pair of tongs cause it's alpha helix connection is so flexible that Calmodulin wraps around its target proteins in a very efficient and flexible manner irrespective of various shapes and/or sizes of the proteins. [5][6][7][10].

Ca^{2+}-CaM binds many kinases, phosphatases, signaling proteins, and structural proteins. This causes changes in a wide aspect of internal processes such as neurotransmitter release, muscle contraction, metabolism, apoptosis, inflammation, membrane protein organization, and cytoskeleton movement [10][15]. These processes work in combination for the smooth and seamless flow of human functioning. This makes Calmodulin a very important protein for all living organisms.

Chapter 3

Short Linear Motifs (SLiMs)

3.1 Short Linear Motifs - SLiMs

SLiMs are minimotifs that are existent in every single eukaryotic cell or say, proteomes. SLiMs regulate a considerable amount of the cell's functions, for example, proteolytic cleavage, and post-translational changes. It furthermore helps in interchanging process as binding sites for cell signaling. They are extremely one of a kind as they take into account encoding for short interactions of smaller protein sequences. With such differences in structures of the cell, SLiMs are the most essential component in the development of cells in the past and for present and as well as for what is to come in future. With such high developmental impact, SLiMs are an extremely defenseless and the best strength of the cell for an attacking infection or virus. They can mislead the virus whose end goal is to hijack the whole cell and it's procedures and cause interruptions in the normal developmental flow [2].

3.2 Properties of SLiMs

SLiMs are located in intrinsically disordered regions [4]. However, their interaction with structured proteins or target's secondary structures is induced. The majority of SLiMs

consist of 3 to 10 contiguous amino acids, with an average of just over 6 residues. This attribute of SLiMs which is having a limited number of residues that directly binds the binding partner leads to two major consequences as well. First, only few or even a single mutation can result in the generation of a functional motif. Secondly, further mutations of the residues lead to changes in the affinity and specificity of their interactions [7]. This results in SLiMs having a tendency to evolve and facilitates their proliferation and development [8]. SLiMs have low affinity in the disordered regions of where they are located with their interaction partners. This makes these interactions temporary and limited for a certain time and also reversible. This property of SLiMs is suitable for arbitration of dynamic processes such as cell signaling. In addition, this means that these interactions can be easily regulated by transitional changes that bring about the change in the structural and physiochemical properties of the motif.

These motifs of 3-10 amino acids are considered as short, linear motifs (SLiMs) or also called as mini-motifs [12]. There are some special properties that distinguish SLiMs from motifs. One of them is that SLiMs have the capacity to encode a functional interaction in a short sequence. they are involved in various internal functions such as assisting in the assemble of protein complexes and control the stability of proteins.

3.3 Function of SLiMs

SLiMs function in every tract of the organism. They have an important role in the regulatory functions of the cell, protein-protein interaction and signal transductions. SLiMs are the interaction modules as they are present in the regions of the protein sequences that cause communication with their target partners. The majority of known interaction partners of SLiMs are globular protein domains although they do interact with RNA and lipids too.

Despite its such abundant uses in the cell growth and development, little is known about SLiMs and their evolution is still emerging. However, they are a major component in protein evolution and due to SLiMs, proteins gain crucial regulatory functions. Thus, proteins networks acquire more interactions through these short patterns and these mini motifs are considered the key substrates for transcriptional regulatory evolution.

3.4 Representation of SLiMs

SLiMs or motifs can be represented in different ways. The most common ways to represent a SLiM are:

- Regular expression
- Sequence logo
- Position specific probability matrix (PSPM)
- Block representation

3.4.1 Regular Expression

A regular expression is the easiest and generally utilized to represent a SLiM. It is the continuous appearance of the amnio acid letters of the SLiM. The numerous letters for a single position appears inside the square sections. For example, the regular expression of one SLiM found in *Sodium channel protein type 5 subunit alpha* (Q14524) at position 192 is $[GP]WN[IW][LF]DF$. This SLiM is seven amino acids long. The first position of the SLiM is either G or P, the second position is W, third is N and so on for further positions. However, the regular expression does not provide any significance, frequency and/or information contained in the letter at that particular position. It only displays the occurrence of that amino acid letter at that position.

3.4.2 Sequence Logo

A sequence logo is a graphical representation of a motif. It indicates how well the amino acids are monitored at every position. The higher the quantity of amino acids, the higher the letter will be as the preservation at that position is higher. Diverse amino acids at the same position are scaled by data contained in that amino acid.

Figure 3.1: SLiM found in Q14524 *Sodium channel protein type 5 subunit alpha* at position 193. The sequence logo was generated using MEME [15].

Figure 3.1 represents the sequence logo for the SLiM found in *Sodium channel protein type 5 subunit alpha* (Q14524) at position 193. The regular expression of the SLiM is $W[NCH][IWM][FLM]D[FS]$.

The x-axis of the sequence logo depicts the positions in the SLiM and the y-axis represents the information contained in a specific position. The information content of the y-axis is given by Equation (3.1):

$$R_i = log_2(20) - (H_i + e_n), \tag{3.1}$$

where H_i is the uncertainty of a position i and H_i is defined as:

$$H_i = -\sum f_{a,i} \times log_2(f_{a,i}). \tag{3.2}$$

Here, $f_{a,i}$ is the relative frequency of amino acid a at position i, e_n is the small-sample correction for an alignment of n letters. The height of letter a in column i is given by Equation (3.3):

$$e_n = \frac{(s-1}{2 \times ln(2) \times n} \tag{3.3}$$

For proteins, the value of s is 20 and n is the length of the SLiM [11].

3.4.3 Position Specific Probability Matrix (PSPM)

A position specific probability matrix (PSPM) is the matrix $\{m_{i,j}\}$ representation of a SLiM. The value of each element represents the probability of the j^{th} letter that occurs at the i^{th} position. Each column in the matrix $\{m_{i,j}\}$ represents each of the 20 amino acids and the rows represent each position of SLiM. Figure 3.2 shows the Position Specfic Probability Matrix for the SLiM found in *Sodium channel protein type 5 subunit alpha* (Q14524).

A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0.2	0	0	0	0	0.2	0	0	0	0	0.6	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0.4	0	0	0.2	0	0	0	0	0	0	0	0	0
0	0	0	0	0.2	0	0	0	0	0.6	0.2	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0

Figure 3.2: Position specific probability matrix of a SLiM found in Q14524 *Sodium channel protein type 5 subunit alpha.*

The regular expression of the SLiM is **W**[**NCH**][**IWM**][**FLM**]**D**[**FS**. The value of the first row and nineteenth column is 1 which states that the value at the first position of the motif is **W** and its probability is 1. The second row has columns second, seventh and twelvth with values 0.2, 0.2 and 0.6 respectively. This means that the value at the second position will be either **C**, **H** or **N** with their probabilities as 0.2, 0.2 and 0.6 respectively. Similarly, the value at the third position will be either **I**, **M** or **W** and the fourth position will contain either **F** or **L** or **M**. The fifth position will hold **D** and the sixth will have the value either **F** or **S**.

3.4.4 Block Representation

One SLiM may show up in numerous destinations (i.e, it may be on the same position or on distinctive positions). The block representation demonstrates the nearness and positions of a SLiM at various positions. The block representation of the SLiMs is

likewise essential to consider the presence of a SLiM in various complex chains and positions.

3.5 Types of SLiMs

There are mainly three types of SLiMs and this structural categorization is based on their different properties. The types of SLiMs are:

- Sequence SLiMs
- Structural SLiMs
- Functional SLiMs

Sequence SLiMs can be characterized as an amino acid sequence design which is spread across the datset. A sequence SLiM is the least complex type of a motif. For a particular site, all the amino acids in the sequence motif originate from a solitary chain of a complex. Figure 3.3 displays the sequence SLiM which was found in Crystal structure of human ryanodine receptor 3 starting from the 194th amino acid up to 200th amino acid covering all the consecutive amino acids.

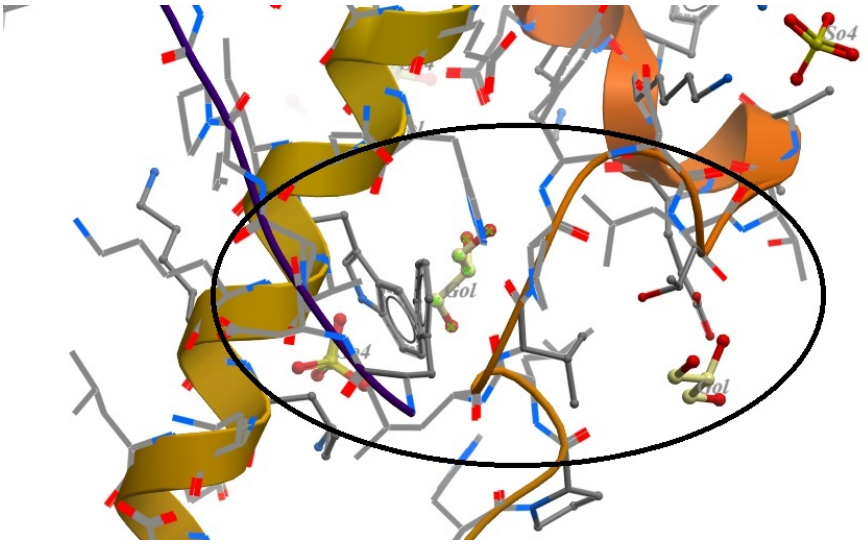

Figure 3.3: Sequence motif of protein with PDB 4erv. This image was created using the ICM browser from the PDB file.

A structural motif is characterized as a blend or combination of amino acids from various parts of a chain or distinctive chains. Thus, they are firm enough to form in to a defined structure. These motifs are naturally more imperative as the structure is preserved and can perform a particular errand task as part of the protein complex [11]. Figure 3.4 represents the structural motif of Crystal structure of the C2H2-type zinc finger domain of human ubi-d4.

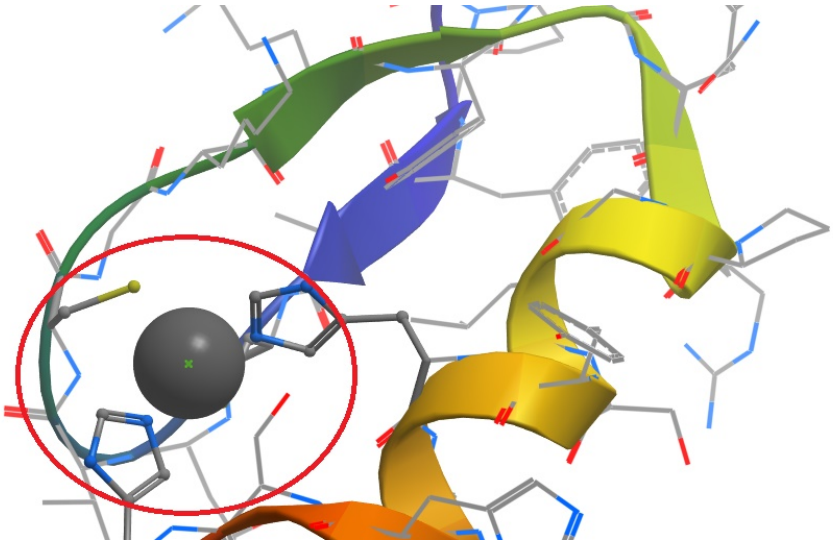

Figure 3.4: Structural motif of protein with PDB 3iuf. This image was created using the ICM browser from the PDB file.

There is another type of SLiM, known as functional motif. They do not necessarily have sequential nor structural formation. There is not much known about these functional motifs. They do perform certain biological activities from their binding sites.

Thus, SLiMs or the mini motifs have great importance in the biological aspect of organisms that many computational methods are evolving to identify new Short Linear Motifs from the protein sequences. Thus, it is considered a major contribution towards the study of proteins.

Chapter 4

Tools for finding Motifs

4.1 Introduction

In the study of proteins, protein structure and protein types, SLiMs are crucial and play an important role in cell organization. Research is currently being conducted on protein sequences, motifs, calmodulin and thus many new databases, tools and software have been developed to facilitate research on Calmodulin. These online tools allow the users to find new motifs or existing motif families for various proteins. These tools have features of both web-based as well as server installation.

4.2 Various Approaches for Motif Finding

Short Linear Motifs are short sequences of a certain patterns found in sequences of DNA or proteins. The discovery of motifs (also called motif finding) is summarized as the problem of finding these short linear motifs in the protein or nucleotide sequences [34]. The discovery of regulatory elements such as transcription factor binding sites (TFBS) in various protein or nucleotide sequences using the motifs is currently the most widely area of research in biology. However, prediction of CaM-binding proteins using these short motifs has not been considered broadly as a research topic. Motif finding tools

are being used to identify the elements in the protein sequences. These tools have not been used for prediction of Calmodulin-binding sites to identify CaM-binding proteins.

The field of computational biology currently suffers from the motif search problem. Scientists consider this problem to be NP-complete [15]. The problem stated is "finding a substring of length l - which is called the motif - that occurs in a set of input sequences $s(s_1, s_2, s_3,, s_t)$" [37].

The authors Tran et al, [14] state that there two major categories for finding motifs: i) General approaches and ii) Web tools for finding motifs. The general approaches are profiles, consensus, projection, graph representations, clustering and tree-based.The Web tools include MEME, GLAM2, CisFinder, W-CHIP Motifs and so on.

The authors of [14] state that implementation of motif finding Web tools consists of two categories. The first category is sequence implementation wherein existing tools are embedded into a Web tool or Web service. The second category involves implementing new algorithms into a Web tool or Web service. Motif finding Web tools provide the facility to upload input sequences of DNA, proteins, or binding sites. The users can then modify or customize their search by changing the default parameters before finally submitting the request. The users are provided with the feature of their results being displayed on the browser or to download them into their system for further study. However, different motif finding Web tools provide different specifications for finding motifs and yield different result formats. Some Web tools provide the options for users to select the database for their search while others directly use their pre-defined database.

There are various tools available to identify the motifs and patterns. Some of them that are currently being used are described below

- SLiMScape
- SLiMFinder
- ELM
- Motif Scan
- Minimotif Miner (MnM)
- SLiMSearch

- ScanProsite
- MEME
- GLAM2

4.2.1 SLiMScape

SLiMScape [42] is short linear motif analysis plugin for Cytoscape, which is an independent tool to visualize protein networks. This tool does not have a streamlined way to visualize motifs in the protein networks. Theory Brien et al. [42] have developed a plugin which discovers new motifs and also searches the known motifs in the protein sequences. The authors state that SLiMScape uses the concept of SLiMFinder to identify the new motifs present within the protein sequences by forming subsets of protein interaction networks. In this plugin, the known motifs are located through matching of regular expressions.

4.2.2 SLiMFinder

Haslam et al. [41] have developed a new tool for discovering motifs (SLiMFinder - Short Linear Motifs Finder). It is a Web server that allows researchers to discover Short Linear Motifs (SLiMs) in the sequences. The authors state that SLiMFinder is a probabilistic SLiM discovery model based on the principles of the SLiMDisc algorithm [37]. The algorithm takes the protein dataset with familiar binding partners as the input. The motifs are then identified by the combination of dimers into long patterns and retain only specific motifs that occur in large number unrelated proteins. This process leads to the identification of motifs with fixed positions of amino acids and later combined to incorporate variable-length wildcards.

The SLiMFinder Web server tool masks the input sequences according to the specifications of the users and identifies the recurring motifs using the SLiMBuild algorithm. This tool requires at least three sequences for the analysis purpose and two parameters i.e., the Uniprot IDs and user-constructed sequence files (Uniprot flat files and FASTA sequences). The job run time varies based on the size or complexity of the input data. The tool returns upto 100 motifs by default with links providing the detailed results of the motif search.

4.2.3 ELM

The Eukaryotic is a manually curated database developed by Dinkel et al Linear Motif (ELM) [36]. The authors claim that the ELM tool contains more than 240 different motif classes with more than 2,700 experimentally validated instances that have been manually curated from more than 2,400 publications.

ELM offers two services: a) database of short linear motif annotation and b) a tool to use this information to discover possible instances of motifs in the given protein sequences. Their database consists of annotations of motif classes described by unique regular expression patterns. The given protein sequence as input is analyzed to match any of the existing regular expression patterns.

The tool accepts two inputs: i) the Uniprot ID and ii) FASTA sequences. The run time of the job execution is a few seconds. The output page results in various subsections of the results providing a user-friendly environment for easy analysis and study.

4.2.4 Minomotif Miner (MnM)

Minimotif Miner (MnM) [35] analyzes the protein sequences for the presence of SLiMs. The authors Tian Mi et al. [35] describe that the model includes 28 attributes of a minimotif. The MnM algorithm ranks the minimotif predictions in descending order of sequence complexity. It also calculates the scoring metric for the location of the minimotif in the protein sequences. The minimotif predictions are filtered between the source and target proteins. MnM uses six external databases that contain more than 300,000 non-redundant protein-protein interactions.

The MnM tool identifies new functions in proteins, determinants of minimotifs in the protein-protein interactions. The result page displays the different subsets of minimotifs and these results can be filtered to allow users to analyze their area of research.

4.2.5 SLiMSearch

SLiMSearch (Short Linear Motif Search) [40] is Web server tool developed by Haslam et al. that allows researchers to find new occurrences of SLiMs in the set of protein sequences. The authors state that the SLiMSearch algorithm performs motif regular expression search in the given set of protein sequences as input. It discovers the overlapping motifs in the sequences and calculates the global and local motif statistics and attributes.

The SLiMSearch algorithm uses its pre-computed databases for calculations of motif attributes. It also calculates the IUPred disorder score whcih is the mean disorder score across the defined residues. The algorithm also calculates the enrichment score for the input motif against the reverse of the motif and randomly shuffled motif.

The SLiMSearch 2.0 server performs quick pattern matching search to locate all the occurrences of motifs in the protein sequences. The pre-computed databases extract the scores and sequence features for each motif occurrence and then the enrichment scores are calculated. The results are interactive and allow users easy explorations and analysis of the returned occurrences of motifs. The tool takes the motif pattern to be compared against the protein dataset as its input. The job executions vary in time according to the input size and complexity. The tool outputs the results in a tabular format containing the motif instances, conservation and disorder statistics, overlapping features and modifications, and overlapping validated motifs.

4.2.6 ScanProsite

PROSITE is a SLiM finding tool developed by Sigrist et al who state that it consists of documentation entries of protein domains, families, functional sites, associated patterns and profiles to identify the patterns. It is a resource for identifying and annotating conserved regions (motifs) in protein sequences. These regions are identified using weight matrices and regular expressions. The weight matrices describe the protein families and modular protein domains and patterns while the regular expressions describe the short linear motifs.

The search options used in the algorithm allow users to find specific combinations of signatures. This feature allows users to search for domains associated with the use SLiMs for a given set of protein sequences. The authors also mention that the search results can be filtered for users to ease their search criteria. For example, users can limit

the results to proteins with particular names, or proteins present in certain tissues, or proteins having a certain size or within a size range.

PROSITE is a tool for identification and annotation of conserved regions in the protein sequences addressing protein families, domain and motifs.

4.2.7 MEME

MEME (Multiple EM for Motif Elicitation) [15] is a Web service available on MEME suite [23]. It can be downloaded and installed locally. The authors Tran et al. [14] describe MEME to be an efficient motif finding tool that is designed for finding non-gap motifs in DNA or protein sequences. MEME tool accepts $\leq$ 60,000 characters in the input file and they all must be in FASTA format. It provides the feature to eliminate the duplicate sequences and sequences with low information that are irrelevant to the motif search. MEME provides the flexibility to users to specify the length of the motif and the number of motifs they require, the number of sites for each motif and number of occurrences of each motif and so on. MEME requires the users to specify the occurrences of the motifs distributed among the sequences, for instance, zero or one per sequence or any number of repititions. MEME provides the option of email address for notifying the results. It does not either create an account or store the results on the server.

The authors of [15] explain about the three different output formats provided by MEME. They are: HTML, XML, and text. The output shows the motifs as local multiple alignments of the input sequences. For every motif, MEME outputs its E-value, number of sites found, motif logos, motif blocks format, motif block diagrams, position-specific scoring matrix, and position-specific probability matrix [20].

The authors of [15] also explain the algorithm used by MEME: the expectation maximization (EM) algorithm [50]. They state that the EM algorithm for motif finding presented by Lawrence et al. [40] has certain disadvantages. The users have no prior knowledge about the selection of the starting position and when to halt finding new motifs. However, MEME resolves this issue and selects the starting points based on all sub-sequences of sequences in the training dataset. MEME removes the occurrences of a motif after its discovery and continues its search for new motifs.

As MEME deletes previously discovered motifs when it searches for new motifs, MEME models a single motif at a time and it does not detect alternative binding motifs.

The user needs to input a set of protein sequences in FASTA format to find new

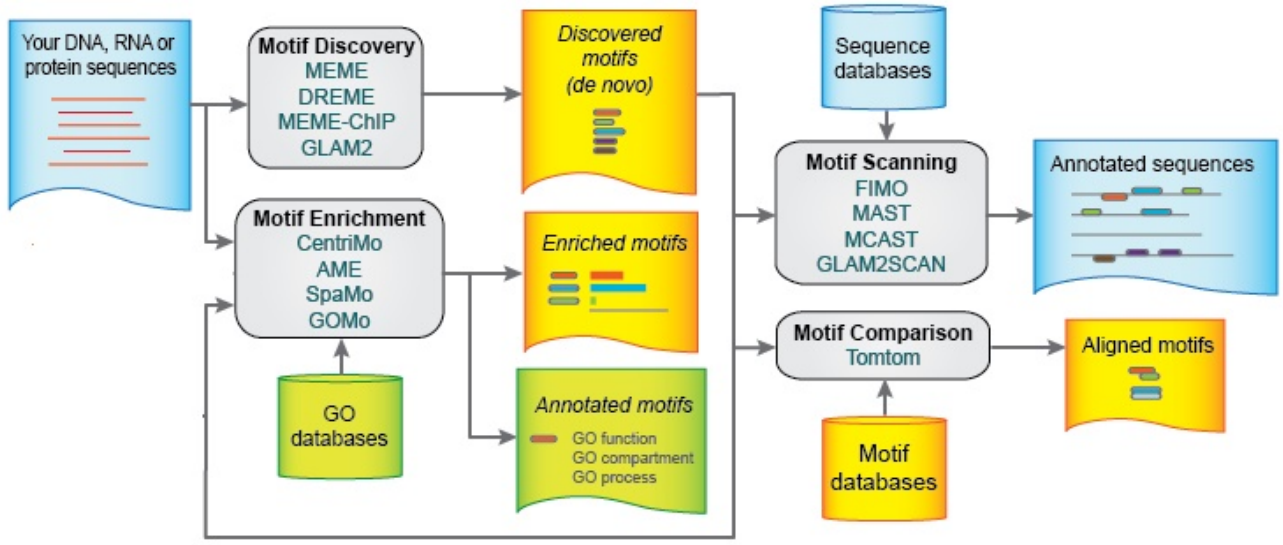

Figure 4.1: Block diagram of MEME software. The diagram was downloaded from http://meme.ebi.edu.au/meme/doc/overview.html [23].

motifs in MEME. The users must either provide the sequence or upload a sequence file. There are various parameters that can improve the results such as: width of the motif, number of motifs, number of sites, and model (ZOOPS, OOPS, ANR).

4.3 Why choose MEME?

There are few SLiM discovering tools accessible to discover SLiMs, for example, SLiMFinder, SLiMDisc, SLiMSearch, MnM, and MEME. However, none of them discovers SLiMs through an unsupervised approach. Also, most of these tools have pre-computed databases from which they discover the short motifs. It is not noteworthy that their databases may not be updated and as such there are chances of missing certain motif discovery. In this thesis, I aim to discover unknown SLiMs in a predefined protein dataset, in an unsupervised manner without the usage of pre-defined databases and so MEME was found to be the best alternative. The MEME suite offers many other options accompanying to it:

- Motif Discovery Tools: MEME, DREME (DNA only), GLAM2.

- Sequence Searches: FIMO, MCAST, GLAM2SCAN.
- Motif comparison tools.
- Motif enrichment tools using SpaMo, CentriMo.

Additionally, MEME depends on the MM calculation which combines the expectation maximization (EM) procedure for pre-defined datasets. For a given dataset, it finds the probability estimation of the parameters by using EM.

The MM algorithm has two advantages over the Gibbs sampling algorithm, proposed by Bailey et al. They state that MM does not need the input sequences to be classified in advance. MM estimates the occurrence of SLiMs from a given data model. This capability is quite robust since the authors claim that even if 20% of the protein sequences in the dataset contain a motif, then that motif can be used for classification and it would classify quite well. Second, the algorithm does the maximum likelihood estimation. This means that its objective is to identify those parameter values that maximize the likelihood of the dataset.

Thus, MEME turns out to be a very efficient and successful algorithm for discovering new SLiMs with different number of occurrences in a set of protein sequences.

Chapter 5

Pattern Recognition

Feature Extraction and Classification

5.1 Features

Classification is performed by utilizing distinctive machine learning and pattern recognition methods. In any case, the principle inquiry is what data that ought to be the input for those machine learning and pattern recognition methods to perform the classification precisely. The properties or data of any object to train a dataset are known as *features*. All the properties of an object are not considered as features, rather only those which are recognizable and are able to identify the object uniquely. In the field of machine learning and pattern recognition, one of the real difficulties is to choose and utilize the right features for effective classification and/or prediction. Mostly, these are numerical values; however, different data types, for example, strings or charts / graphs can be taken into consideration as features.

5.1.1 Features used to predict CaM-Binding proteins

The Short Linear Motifs are used as features in my thesis which helps in classifying the dataset of proteins as CaM-Binding and non CaM-Binding. The SliMs are a list of known CaM-binding motif families recorded in [9] and unknown new motifs discovered by MEME [67].

Known Motifs

The family of known motifs are taken from Mruk et al [9]. The known motifs comprise those listed in Figure 5.1.

Known Motif Family	Sequence
1_10	[FILVW]xxxxxxxx[FILVW]
1_15_10	[FILVW]xxx[FAILVW]xxxx[FILVW]
Basic 1_5_10	[RK][RK][RK][FAILVW]xxx[FILV]xxxx[FILVW]
1_12	[FILVW]xxxxxxxxxxxx[FILVW]
1_14	[FILVW]xxxxxxxxxxxx[FILVW]
1_8_14	[FILVW]xxxxxx[FAILVW]xxxxx[FILVW]
1_5_8_14	[FILVW]xxx[FAILVW]xx[FAILVW]xxxxx[FILVW]
Basic 1_8_14	[RK][RK][RK][FILVW]xxxxxx[FAILVW]xxxxx[FILVW]
1_16	[FILVW]xxxxxxxxxxxxxx[FILVW]
IQ	[FILV]Qxxx[RK]Gxxx[RK]xx[FILVWY]
IQ-like	[FILV]Qxxx[RK]xxxxxxxx
IQ-2A	[IVL]QxxxRxxxx[VL][KR]xW
IQ-2B	[IL]QxxCxxxxKxRxW
IQ-unconventional	[IVL]QxxxRxxxx[RK]xx[FILVWY]

Figure 5.1: Known motifs retrieved from Mruk et al. [9].

New Motifs

The new motifs have been discovered from MEME [23]. It has provided a hundred motifs through its MM algorithm. The new motifs are those listed in Figure 5.1.

Table 5.1: New Motifs.

Motifs	Sequence
1	[GP]WN[IW][LF]DF
2	N[DT]N[SG]S[RQ]F
3	CTI[KL]T[DN][WC]
4	NYH[IV]FYQ
5	IPSFMCQ
6	PH[FY][VI]RCI
7	DLKPEN[IL]
8	EQ[FL]CIN[YF]
9	[VT]FTGI[FY]T
10	[ST]T[FW]HR[IV]
11	PPPPPPP
12	FRALCTG
13	LDI[YAD]GF[EG]
14	GDFTNHN
15	TNEMWRS
16	H[VI][VD]FGHV
17	YLLEKSR
18	E[AG]F[GR]NAK
19	RDVAYQY
20	[FL]I[FY][AS][IV][IV]G
21	[FT]K[TC]F[GI]CG
22	YYN[EA]MK
23	GW[MT]D[IV][ML]Y
24	VITDCGQ
25	Y[MT]DKNNV
26	YGSRFPD
27	IYTY[SI]G
28	KCP[PQ][YC]W
29	VDHYENP
30	TIIADNI
31	K[HN]VG[PT]G[VL]
32	KGFGYKG
33	PARLYHK
34	P[DHN]DY[NF]CK
35	MDVVKKI
36	SMANAGP

Motifs	Sequences
37	TMWDCM
38	KNPVTIF
39	W[IM]VTRIN
40	CGDVMKL
41	QDPV[ML]RQ
42	PPVKLHC
43	CPICR[WD][NQ]
44	I[WL]DTAGQ
45	EWVKGKT
46	CCW[HIT]C[ET]
47	YQ[ML]GDVS
48	[GA][VL]I[IM]D[NAT]F
49	NPLVYLD
50	FFNHHMF
51	HS][VI]HNFM[MA]
52	HRWYP[HR]G
53	CW[IW][ET]IH[LP]
54	TIKNTDI
55	GCCCR[YD]
56	YCR[VL]WRW
57	DFWRM[IV]W
58	C[IV]NPYHY
59	VK[HN]WPWM
60	FTMYTTC
61	I[RC]HWIGC
62	[ML]DPY[YE]YF
63	ADVVPKT
64	MFRRPV
65	MRMADFW
66	[HMN][NF][NW]FETF
67	IVCYHP
68	MWGRNVY
69	CSTCH
70	QQQQQQ
71	[GS]LW[WL]TY
72	QNFDYMF
73	HNGCRP
74	QFVRHES

Motifs	Sequences
75	CP[NH][RIT]V[DA]C
76	ERCW[KD][FY][FY]
77	EGK[VI]DYG
78	PEQRGMF
79	[EQ]W[KR][QT]KYE
80	LFW[SG]IF[GD]
81	GTGGKSI
82	[FY]F[VI]I[YF]II
83	EYV[YT]KGQ
84	SVFRPGL
85	[TS]IGYGD[KM]
86	[LP]TEY[CI][QH]G
87	GSF[FY][LT][LI]N
88	L[VC]GN[KS][SCT][DE]
89	EHN[LM][AW][NH]Y
90	K[TI]YMIFF
91	VTGD[GN][IT]N
92	CTNRPFY
93	MH[YF]GNMK
94	NMVTMMV
95	[SA]GAGK[TS]
96	RT[FL]R[VL]LR
97	RT[FL]R[VL]LR
98	GT[PV]GY[ML]A
99	[LC][VI][TA]VNPY
100	DMWSIGV

5.2 Feature Selection

Feature selection is the process of choosing the best subset of relevant and critical features from the entire feature set. This process selects the most relevant features that represent the whole group of features efficiently after eliminating the redundant or irrelevant ones. Applying feature selection before executing a classifier is useful in reducing the dimension of the data. This leads to reduction in the prediction time thereby improving on the prediction performance.

There are two ways of performing feature selection:

- wrapper methods and
- filter methods

In filter-based feature selection method, the quality of the selected features are scored and ranked independently of the classification algorithm. This is done by using certain criteria based on their relevance. The feature selection based on filter methods is quick but it does not consider the dependency of features from each other. This means that a particular feature is not useful in itself but can be very useful when combined with others. Some of the most renowned filter methods are the Minimum Redundancy Maximum Relevance (mRMR), Information Gain (IG), Gain Ratio (GR) and Chi Square (x^2).

On the other hand, wrapper methods find the best subset of features using a particular predictive model (classifier) to score feature subsets. Performing exhaustive search executions to find the best subset of features is computationally intensive. Some heuristic search methods can be employed to find a subset of optimal features for a specific dataset such as forward selection and backward elimination.

5.3 Classification

After feature extraction and selection of the most discriminating features, a classifier is applied to assign the class labels (Prediction of CaM-binding proteins). For this, the samples or dataset are first divided into train and test sets using different methods such as m-fold cross-validation or leave-one-out method.

The classification method follows two phases of processing for training and testing. In the training phase, the training samples are used to develop a model that is a description of each training class. Then, in the testing phase, that model is used to predict the classes of the test samples.

The classification comprises deciding the class of an unknown item (the class to which the object belongs to). There are predominantly two diverse types of classification; supervised and unsupervised classification. In both categories, classification is performed on the premise of the given training samples. For supervised

classification or grouping, the training dataset, is given labels and if there should be an occurrence of unsupervised order, the training dataset is left with no labels. With the given training dataset, the classifiers perform the classification and can classify an unknown sample. As the training dataset of an unsupervised classifier does not have label data, it endeavours to locate the characteristic samples in the information that can ideally decide the right class for the unknown element.

There are different types of classifiers available:

- Linear classifiers
- Support vector machine
- Nearest neighbour
- Decision tree
- Neural networks
- Bayesian Classifiers
- Random Forest

Of these commonly used methods, in the thesis I use support vector machines (SVMs), k-nearest neighbor (k-NN) and Naive Bayes for the prediction of CaM-binding proteins.

5.3.1 Support Vector Machine

I have utilized SVM, which is generally utilized in bioinformatics problems for classification purposes. It takes the arrangement of input vectors and predicts the conceivable classes of output in view of the support vectors. The kernel trick permits to outline unique vectors onto a much higher dimensional space that ideally makes class division less demanding [11].

The way SVM functions relies upon the training information or dataset. If the training data is linearly separable then the distance between the classes of data can be increased to the maximum. This method is applied by selecting two hyperplanes parallel to each other. The region formed between the two selected hyperplanes is called

margin. Margins are of two types: soft and hard margin. In the hard margin SVM, the training information is linearly separable in the data space. In case the training dataset is not linearly separable then they are mapped onto a higher dimension feature space to expand the distinctiveness. This process is termed as soft margin.

5.3.2 *k*-Nearest Neighbor

For classification and data mining, *k*-nearest neighbor is one of the main classifier on account of its effortlessness. The *k*-NN technique utilizes the concept of Cicero pares cum paribus facillime congregantur (birds of a feather flock together) [70]. It classifies an unknown specimen sample based on the known sample of its neighbour. It calculates the distance between the test sample and the training samples. The distance with the smallest value corresponds to the sample in the training set nearest to the unknown sample, and the unknown sample is delegated the class of that training sample.

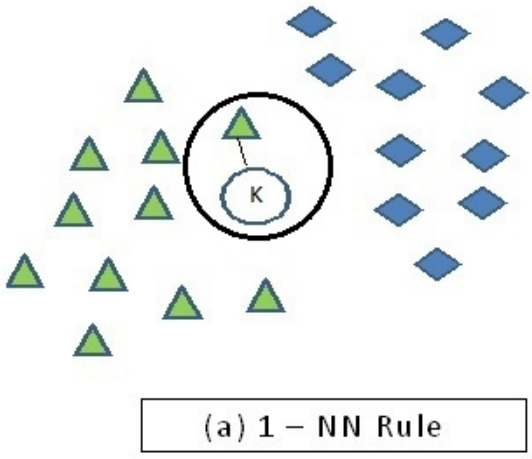

Figure 5.2: Schematic view of *k*-NN classification for k=1.

In Figure 5.2, the 1-NN rule is applied as only one instance is used for classification from the training samples. The unknown sample is calculated only once.

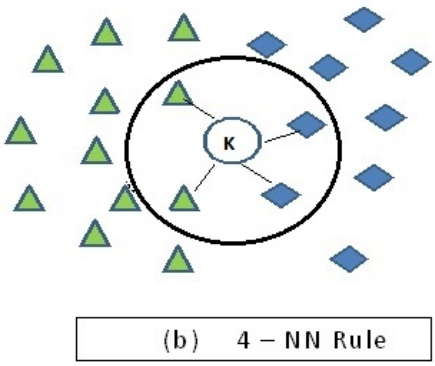

Figure 5.3: Schematic view of k-NN classification for k=4.

Figure 5.3 represents the conventional view of the k-NN classification. In this 4-NN rule, the unknown sample is compared with the 4 training samples and the class with the largest number of matching sample is selected for the prediction.

The k-NN classification requires two points to be considered while selecting it for classification. One is the distance function and the other is the estimation of k. k-NN considers the distance function so that the shorter the distance is, the more prominent is the probability of the sample to belong to that class. The second significant point is the estimation of k, which represents the count of the closest neighbors for the unknown sample. If the estimation of k is too substantial, classes with the expansive number of classified samples will overpower smaller ones. On the contrary, if the estimation of k is too small, the benefit of utilizing numerous samples as a part of the training set is not violated. Normally, the estimation of k is enhanced by trial and error on the training and validation sets.

The k-NN method does not produce an apt classifier using the training samples, but it uses the training samples each time it classifies an unknown sample. Hence it is called lazy classification method, which is much easier but computationally expensive.

5.3.3 Naive Bayes

Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with independent assumptions between the features. It is a known method for text categorization, the problem of investigating documents belonging to a particular category or the other (such as spam or true) with word frequencies as the features [72].

Naive Bayes is highly scalable and requires a number of parameters to be linear to the number of features in a given dataset. Training is done by evaluating a closed-form expression, which takes linear time, as compared to other classifiers that use expensive iterative computations.

In Naive Bayes classifier, the class assigns labels to dataset samples and is represented as vectors of feature values. The class labels are drawn from some finite set. Naive Bayes classifier is trained very efficiently in a supervised learning area. In many practical applications, parameter estimation for Naive Bayes models uses the maximum likelihood method.

An advantage of Naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification. This makes Naive Bayes a very simple and efficient classifier for prediction and classification analysis.

In this probability model, a given problem instance to be classified is represented by a vector $x = (x_1, ..., x_n)$ representing n features. The conditional probability is represented as:

$$p(C_k|x) = \frac{p(C_k)p(x|C_k}{p(x)} \tag{5.1}$$

Equation 5.1 can be expressed in simple terminology as:

$$posterior = \frac{prior \times likelihood}{evidence} \tag{5.2}$$

5.4 Classifier Evaluation

To validate a classifier, the result needs to be validated. There are different methods available for validation. I briefly summarize the most widely used ones.

5.4.1 Cross validation

Cross validation is an evaluation method for independent data models. This method does not use the entire dataset for training. It separates the dataset into two subsets: training set and test set or validation set. During the training phase, some part of the dataset is removed. After the completion of the training phase, this removed section of the dataset is used to test the performance of the resulting model on the new data. There are three cross-validation methods:

- m-fold cross validation
- Holdout method, and
- Leave-one-out cross validation

The most widely used method is m-fold cross validation. In my thesis, I have used two-fold cross validation to validate the results.

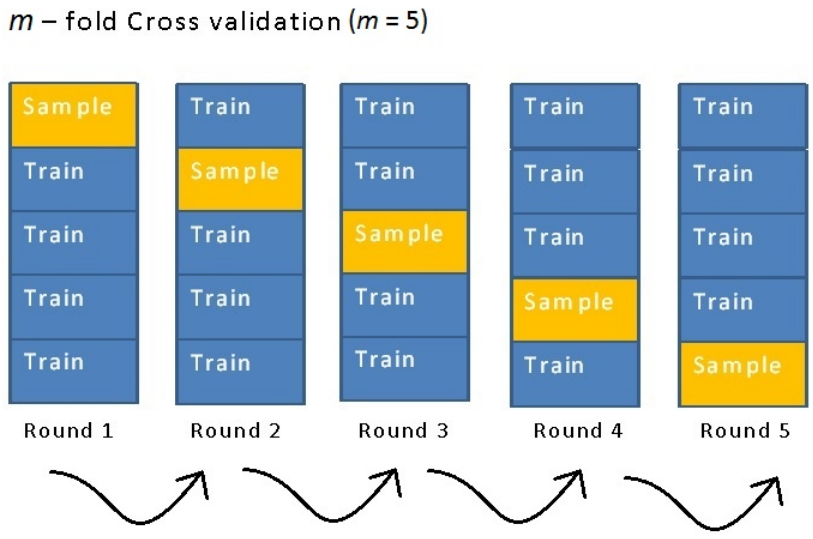

Figure 5.4: Schematic view of 5 - fold Cross Validation.

Figure 5.4 depicts the 5-fold cross validation graphically. The entire dataset is divided into five parts: one part (orange) is used for test and the other four parts (blue) are used for training purposes. After each iteration, the test and training datasets are swapped and in this manner the whole dataset is covered and validated.

The total values for TP, TN, FP and FN are computed to calculate the accuracy as stated in Equation 5.3. To compute the accuracy for each classifier, the following equation is used:

$$Accuracy = \left\langle \frac{TP + TN}{TP + FP + TN + FN} \right\rangle \times 100\% \tag{5.3}$$

where:

- TP - when positive is classified as positive
- TN - when negative is classified as negative

- FP - when negative is classified as positive
- FN - when positive is classified as negative

For unbalanced class problems, the performance can be analyzed in terms of specificity (SP = TN=N), sensitivity (SN = TP=P) where,

- P - Number of CaM-binding proteins in positive class
- N - Number of CaM-binding proteins in negative class

TP is the number of correctly identified CaM-binding proteins and TN is the number of correctly identified non-CaM binding proteins. FP and FN are numbers of incorrectly classified CaM-binding and non CaM-binding proteins respectively.

Also, the receiver operating characteristic (ROC) curve is a visual tool that is plotted based on the true positive rate (TPR) or also called sensitivity, vs. the false positive rate (FPR), or also called "specificity. To generate the ROC curves, the sensitivity and specificity of each subset of features are determined for different parameter values of the classifier. Then, by applying any simple classification algorithm, the FPR and TPR values are filtered as follows:

- for the same FPR values, the largest TPR value is selected, and
- for the same TPR values, the smallest FPR value is selelcted.

A polynomial of degree 2 is then fitted to the selected points. ROC analysis is suitable for unbalanced class problems and yields a better insight than simple performance metrics.

5.4.2 Holdout Method

This method separates the dataset into training and test sets. This is the easy version of *m*-fold cross validation. The training set is used to train the classifier and the test set estimates the error rate of the dataset that was trained by the classifier. However, this method has certain disadvantages: a) If the dataset is sparse then dividing the test and training sets will be costly for performing the classification. b) The cross validation is executed one fold and hence there are chances that the error rate would be inaccurate.

5.4.3 Leave-One-Out

In this method, in each iteration, one sample is kept apart and the remaining are used to train the classifiers to classify that sample which was kept apart while training the classifier. As it is an iterative process, it covers all the samples. Therefore, leave-one-out is computationally expensive.

Thus, pattern recognition offers many different classifiers which are used for classification based on the type of dataset. The dataset used for this thesis considers all the efficient pattern recognition classification methods so as to improve upon the prediction performance.

Chapter 6

Implementation

Model Implementation

In this thesis, I have implemented a new model that predicts the CaM-binding proteins using the known and new SLiMs contained in the protein sequences. This chapter discusses the proposed model in detail.

6.1 Dataset

In this proposed model, two different datasets have been used: The dataset of CaM-binding proteins and another dataset non-CaM binding proteins are then combined to form one dataset for both classes.

- Calmodulin-binding Protein Dataset - The proteins have been collected from the *Calmodulin Target Database* [12]. This is a web-based database for the family of proteins that contain Calmodulin-binding sites.
- Non Calmodulin-binding Protein Dataset - These proteins are Mitochondrial proteins collected from *Uniprot* [27]. Although there were more than thousands of

proteins we have manually curated the original dataset and selected 224 proteins which have low regions of Calmodulin-binding.

Table 6.1 displays an overview of the dataset used in the thesis. There are 194 CaM-binding proteins and 224 Non CaM-binding proteins that have been curated.

Dataset	CaM-binding Proteins	Non CaM-binding Proteins
Number of Samples	194	224

Table 6.1: Overview of the proposed dataset.

6.1.1 CaM-Binding Proteins Dataset

The CaM-binding protein dataset is composed of 194 human CaM-binding proteins with seven known motif families and new motifs generated by MEME.

CLASS	PROTEIN ID	M94	M95	M96	M97	M98	M99	M100	M101	M102	M103	M104	M105	M106	M107
1	P35499	1	0	1	0	0	0	0	215	95	195	170	90	52	167
1	P25098	0	0	0	0	0	0	0	50	15	40	38	10	1	36
1	Q02952	0	0	0	0	0	0	0	49	13	39	32	8	4	36
1	P28222	0	0	0	0	0	0	0	39	23	29	34	14	10	32
1	P35626	0	0	0	0	0	0	0	45	12	37	40	8	1	37
1	P01833	0	0	0	0	0	0	0	37	9	44	35	12	4	35
1	P08238	0	0	0	0	0	0	0	31	15	45	30	11	2	40
1	Q92952	0	0	0	0	0	0	0	38	14	41	31	15	7	27
1	P08908	0	0	0	0	0	0	0	38	20	33	30	11	7	33
1	Q14123	0	0	0	0	0	0	0	35	16	35	31	12	4	35
1	Q15835	0	0	0	0	0	0	0	40	16	31	36	9	4	31
1	Q14573	0	0	0	0	0	0	0	194	69	209	184	70	32	183
1	O75712	0	0	0	0	0	0	0	32	20	29	31	18	13	23
1	Q01064	0	0	0	0	0	0	0	44	15	34	30	10	2	28
1	P08034	0	0	0	0	0	0	0	28	15	36	30	16	11	25
1	Q9H4G0	0	0	0	0	0	0	0	37	17	35	33	8	1	28
1	Q9UEY8	0	0	0	0	0	0	0	48	12	30	30	7	2	29
1	P11171	0	0	0	0	0	0	0	34	12	38	35	8	2	29
1	O95377	0	0	0	0	0	0	0	32	15	30	28	16	10	26
1	O95452	0	0	0	0	0	0	0	29	13	30	33	18	10	24
1	P54750	0	0	0	0	0	0	0	31	9	30	34	11	5	34
1	Q16566	0	0	0	0	0	0	0	33	14	32	27	11	4	29
1	Q14524	1	0	1	1	0	0	0	202	92	179	166	76	46	165

Figure 6.1: CaM-Binding proteins and their SLiMs.

Table 6.1, shows some of the CaM-binding proteins used for the experiments from the list of 194 proteins dataset. This table is a partial dataset extracted from the whole dataset used for the experiments. The first column represents the class label, the second is the protein ID and the other columns are all the SLiMs both known and found by MEME.

6.1.2 Non CaM-Binding Dataset

The Non CaM-binding protein dataset is composed of 224 Mitochondrial proteins. They also have the seven known motif families and the new motifs generated by MEME.

CLASS	PROTEIN ID	M94	M95	M96	M97	M98	M99	M100	M101	M102	M103	M104	M105	M106	M107
2	F5H612	0	0	0	0	0	0	0	1	0	1	3	2	0	1
2	Q8IXM3	0	0	0	0	0	0	0	14	5	1	5	2	2	3
2	Q3SZ86	0	0	0	0	0	0	0	7	2	8	3	1	1	10
2	H2R5Z4	0	0	0	0	0	0	0	14	5	1	5	2	2	3
2	D4A040	0	0	0	0	0	0	0	7	1	10	7	0	0	7
2	P82931	0	0	0	0	0	0	0	6	2	7	8	3	1	5
2	Q9BQC6	0	0	0	0	0	0	0	9	5	4	5	3	1	5
2	F7FM50	0	0	0	0	0	0	0	5	0	7	10	4	2	5
2	Q9EPJ3	0	0	0	0	0	0	0	9	1	9	4	1	1	8
2	V9GYJ3	0	0	0	0	0	0	0	10	3	8	7	2	0	3
2	Q14197	0	0	0	0	0	0	0	10	2	7	6	2	1	5
2	A8PU71	0	0	0	0	0	0	0	0	0	5	1	0	0	2
2	D4A7X1	0	0	0	0	0	0	0	5	0	6	10	4	1	7
2	Q63750	0	0	0	0	0	0	0	8	2	8	8	2	0	6
2	F8WF37	0	0	0	0	0	0	0	8	1	7	8	2	2	6
2	Q9Y3D3	0	0	0	0	0	0	0	5	1	6	10	4	1	7
2	Q9NS18	0	0	0	0	0	0	0	9	3	7	4	1	1	9
2	G3X705	0	0	0	0	0	0	0	6	2	8	8	3	1	6
2	Q32PC3	0	0	0	0	0	0	0	11	0	12	6	0	0	6
2	P82912	0	0	0	0	0	0	0	9	4	7	7	2	0	6
2	P10790	0	0	0	0	0	0	0	9	3	7	7	2	0	7
2	Q2TBG7	0	0	0	0	0	0	0	11	3	7	5	1	0	8
2	E9PRD2	0	0	0	0	0	0	0	2	0	1	3	2	0	1
2	Q9BYN8	0	0	0	0	0	0	0	8	2	8	6	1	1	9

Figure 6.2: Non CaM-Binding proteins and their SLiMs.

Table 6.2, shows some of the Non CaM-binding proteins used for the experiments from the list of 224 proteins dataset. This table is also a partial dataset extracted from the entire dataset used for the experiments. The first column states the class label, second is the protein id and the other columns are all the SLiMs both known and found by MEME.

6.2 Tools

In my thesis, I have used two tools to process and parse the SLiMs contained in the protein sequences. They are:

- Java Regex

- MEME Suite

The Java Regex tool identifies the sites of CaM-binding motifs from the list of known SLiM families as well as the new motifs generated by MEME. The MEME suite generates the new SLiMs from the protein sequence dataset input to it.

6.2.1 Java Regex

Java provides the Regex package for pattern matching with regular expressions. A regular expression is a special sequence of characters that describes a regular language and helps match or find other strings or sets of strings, using a specialized syntax held in a pattern. They can be used to search, edit, or manipulate text and data.

The Regex package consists of following three classes:

- Pattern Class: A Pattern object is the representation of a regular expression. To create a pattern, first its public static compile() methods is invoked, which will return a Pattern object. These methods accept a regular expression as the first argument.
- Matcher Class: A Matcher object is the engine that interprets the pattern and performs match operations against an input string. The Matcher object is obtained by invoking the matcher() method on a Pattern object.
- PatternSyntaxException: A PatternSyntaxException object is an unchecked exception that indicates a syntax error in a regular expression pattern.

To identify SLiMs in the protein sequences in the dataset, I have used the text pattern matching via regular expressions. I have used a custom Java code that searches the full sequence of each protein in my dataset for subsequences that matches any of the known and new SLiMs. The matches to motifs are recorded as the frequency of occurrence in each protein.

6.2.2 MEME

The new motifs are discovered by the MEME tool. MEME is explained in detail in Chapter 4. MEME lists out the results in three output files (HTML, XML and text

documents) for user's convenience to choose their files for analysis.

6.2.3 Weka

Waikato Environment for Knowledge Analysis (Weka) is a popular suite of machine learning software written in Java. It was developed at the University of Waikato, New Zealand. It is a free software licensed under the GNU General Public License [26].

Weka is a software that contains a collection of visualization tools and algorithms for data analysis and predictive modeling. It collaborates with graphical user interfaces for easy access to these functions. The advantages of Weka are:

- It is freely available under the GNU General Public License.
- It is portable as its implementation is written in Java. This makes it executable on almost any modern computing platform.
- It offers comprehensive collection of data preprocessing and modeling techniques.
- It provides easy access to users since it offers a user-friendly graphical environment.

Weka provides support to several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. All Weka's techniques are predicated on the assumption that the data is available in one file or relation, where each dataset is distinguished by a fixed number of features [26].

I have used Weka for classification and prediction of CaM-binding proteins. The classifiers used are SVM, k-NN and Naive Bayes. This tool also provides the feature selection module. Through this module I have performed feature selection on the 107 features and selected the most discriminative features for classification.

6.3 Model

In this thesis, I propose a model, to predict CaM-binding proteins using the SLiMs contained in the protein sequences. I have used cross validation to validate the predictions. The model contains the following components:

- Protein Sequence Dataset Compilation
- SLiM finding
- Separating sample and training sequences
- Curating the Dataset
- Prediction Module

Figure 6.3 shows the work flow diagram of the proposed model. The next few sections describe all the components of the model in detail.

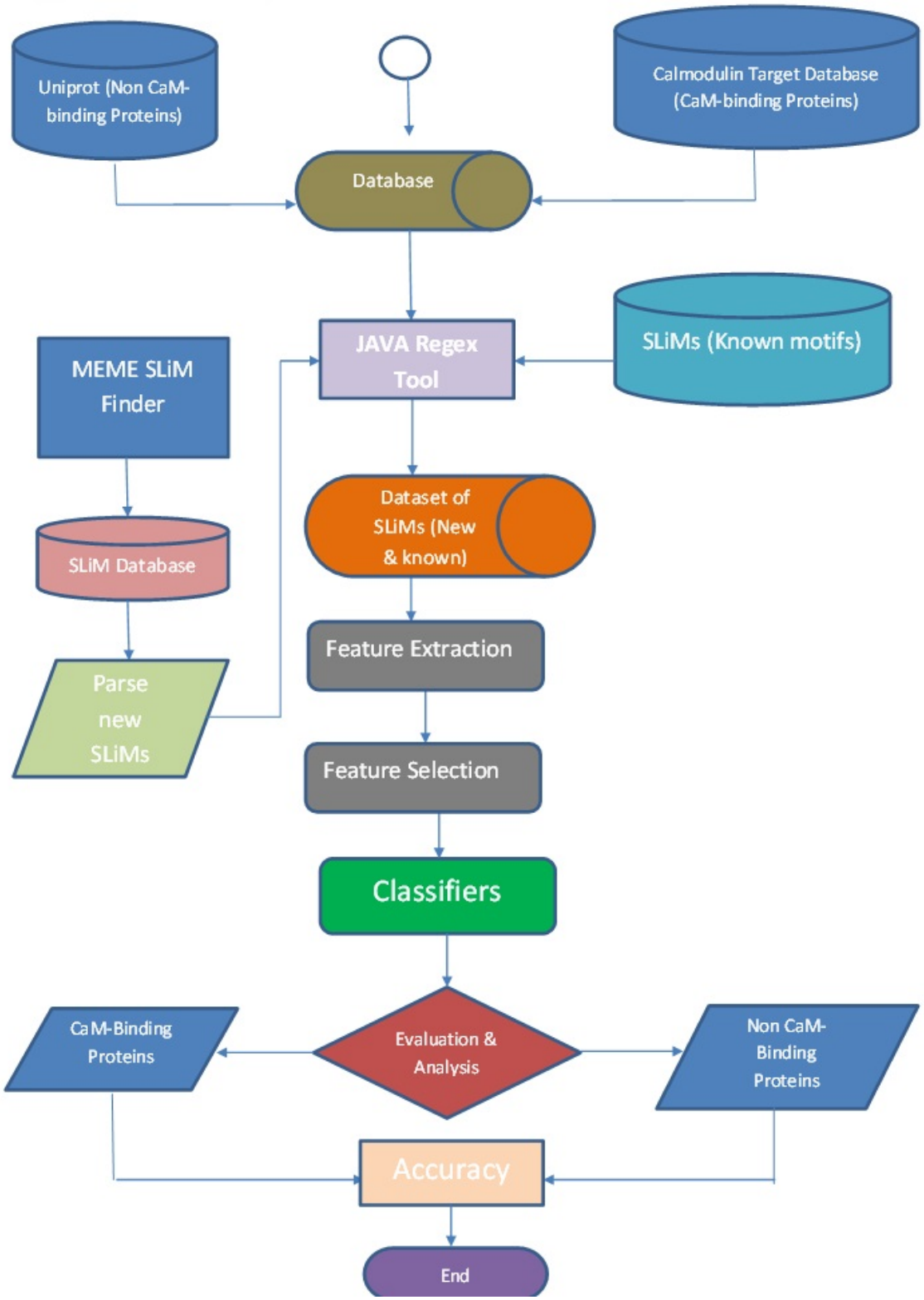

Figure 6.3: Work flow diagram of the proposed model.

6.3.1 Protein Sequence Dataset Compilation

This component takes the list of all CaM-binding and Non CaM-binding protein IDs for a dataset as input. For each protein ID, it downloads the FASTA sequence file from the PDB server. It extracts the FASTA sequences and saves them into a file. This file contains all protein sequences to be used for prediction.

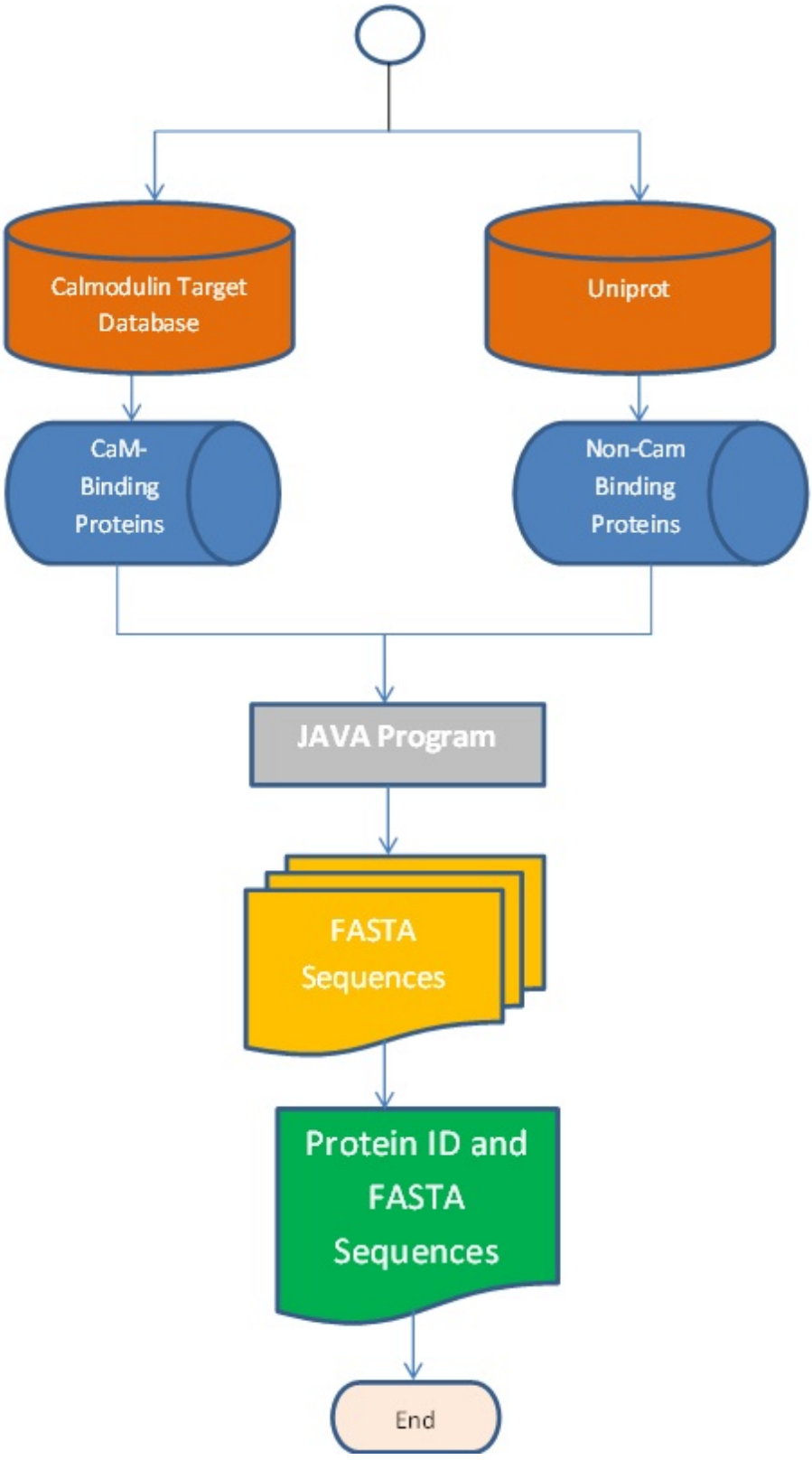

Figure 6.4: Diagram for the data collection approach.

Figure 6.4 depicts the work flow of the collection and compilation of the protein dataset. The list of CaM-binding and non CaM-binding proteins are combined together to fetch their FASTA sequences from Uniprot. The FASTA sequences retrieved are then stored in a file to form the dataset of protein IDs and their sequences.

6.3.2 Short Linear Motif Finding

I have used the known motif family pattern and MEME to find SLiMs from the protein dataset. I have used Java Regex which finds the known motif family pattern in each of the protein sequences and records the frequency of their occurrence in the proteins. I have optimized the parameters of MEME to find the new motifs. I ran MEME to find 80, 100 SLiMs from the dataset. The lengths of the SLiMs were set to 2 - 7.

6.3.3 Separating Training and Test Samples

The motif dataset is divided into target samples and training samples to implement the cross validation process. Each iteration step, the target sample and training samples are separated. As it is an iterative process, the same procedure continues until it covers the entire protein list. The classifier used in the model handles the cross validation.

6.3.4 Feature Extraction Module

The dataset of protein IDs and their sequences is then input to the Java regex tool to parse the SLiM patterns in the protein sequences. This initial dataset is also provided to the MEME suite to generate the list of new motifs. The pattern (motifs) is matched with the protein sequences to find the sub-sequences that match the motif family (both new and known) and recorded as frequency of occurrence. This dataset is compiled by string the protein IDs, the motif occurrence and given class labels. Figure 6.5 depicts the flow of the dataset formation.

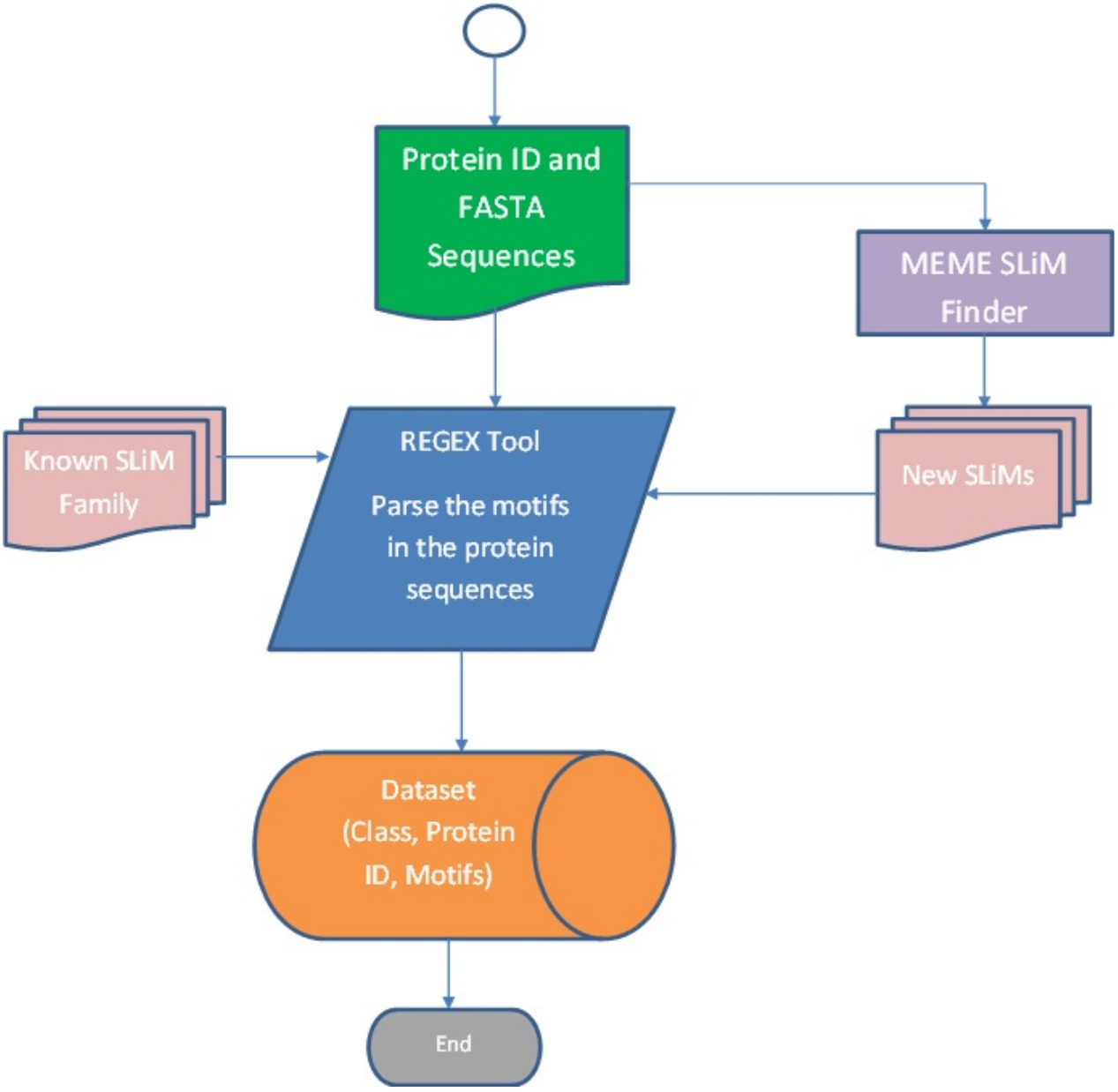

Figure 6.5: Feature extraction module.

6.3.5 Prediction Approach

The curated dataset comprising the class labels, SLiMs as features are provided to Weka for classification and prediction of CaM-binding proteins. The dataset is preprocessed to convert its numeric data. This dataset comprises all the numerical data except the class labels. Accordingly, the feature selection approach is applied to this dataset. I have used ChiSquare feature selection to provide the list of distinguishable and non redundant features. After the feature selection process, the discriminative features are kept in the dataset and others removed. Finally, the classification algorithm is applied to predict the CaM-binding proteins and find the performance metrics. I have used SVM, k-NN and Naive Bayes classifiers to classify the CaM-binding proteins. Figure 6.6 depicts the work flow of the prediction module.

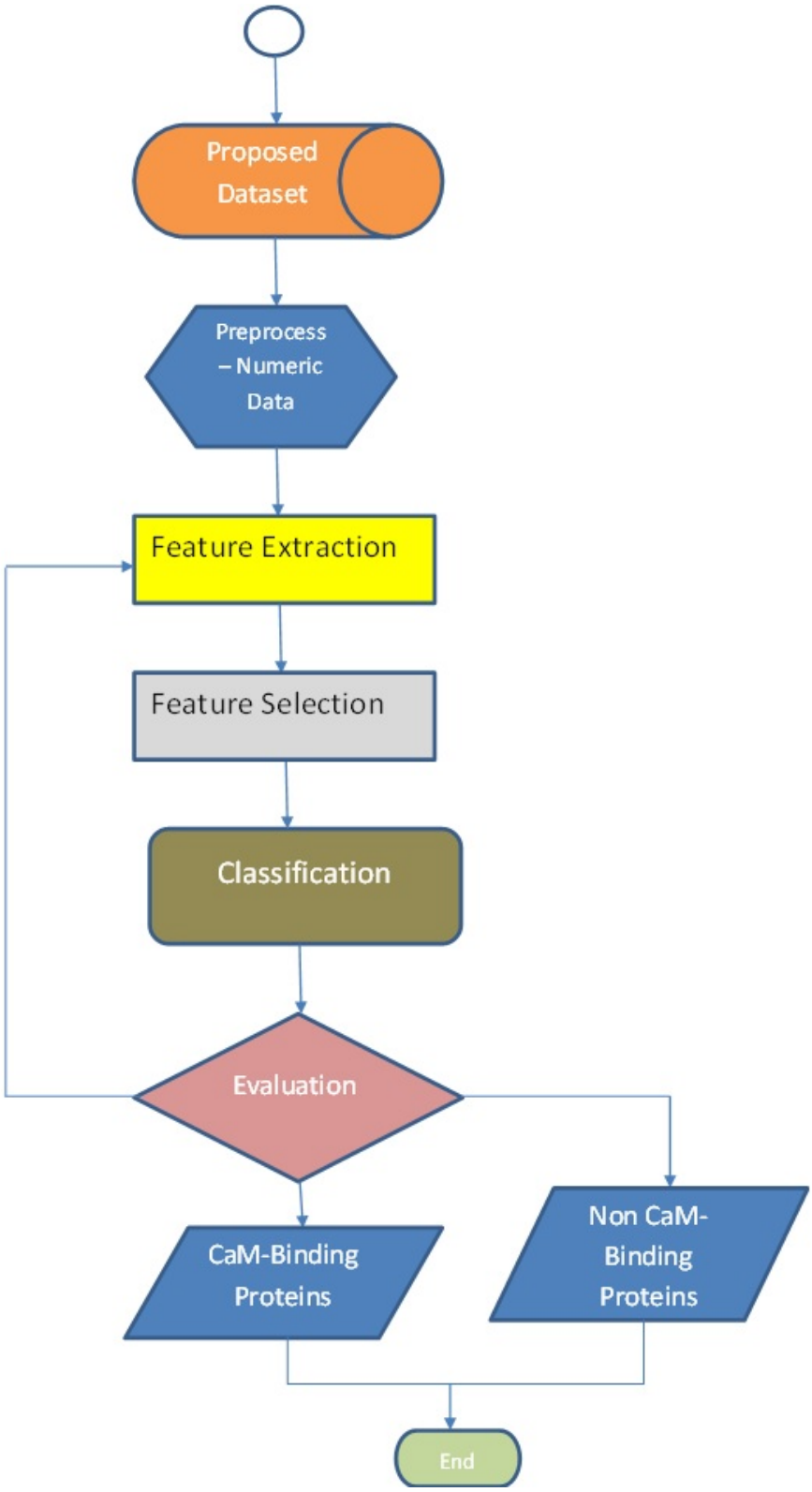

Figure 6.6: Work flow of the prediction module of CaM-binding proteins.

All these components function together accurately and thereby form the proposed model that predicts CaM-binding and Non CaM-binding proteins using SLiMs as features.

Chapter 7

Results and Discussions

To measure the performance and accuracy of the proposed model, different experimental models were adopted. For all the experiments and analysis, the curated dataset has been used. In this chapter, the results for different experimental models are discussed and analyzed.

7.1 Dataset Analysis

As discussed in Chapter 6, the proposed dataset contains 194 Calmodulin-binding proteins and 224 Non Calmodulin-binding proteins or Mitochondrial proteins. The SLiMs used for the dataset are of two categories: Known motif family and new motifs generated by MEME.

The known motif family considered for the curated dataset are provided in Table 7.1.

Motifs	Sequence
1-10	[FILVW][A-Z]8[FILVW]
1-5-10	[FILVW][A-Z]3[FAILVW][A-Z]4[FILVW]
1-12	[FILVW][A-Z]10[FILVW]
1-14	[FILVW][A-Z]12[FILVW]
1-8-14	[FILVW][A-Z]6[FAILVW][A-Z]5[FILVW]
1-5-8-14	[FILVW][A-Z]3[FAILVW][A-Z]2[FAILVW]
	[A-Z]5[FILVW]
1-16	[FILVW][A-Z]14[FILVW]

Table 7.1: Motif family used in the classification experiments.

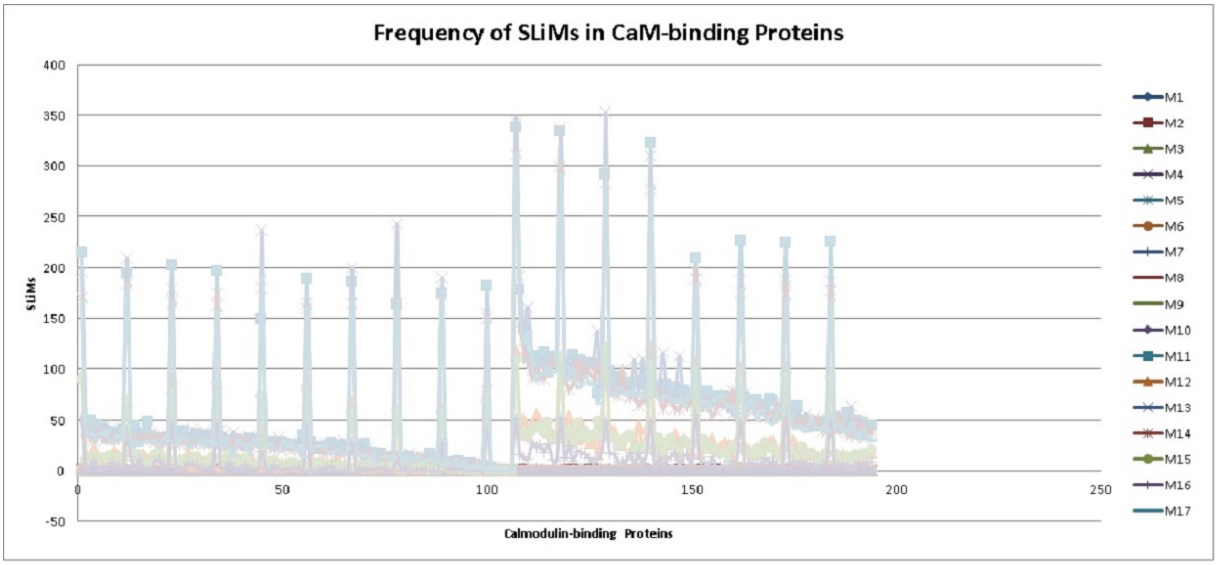

Figure 7.1: SLiM Frequency occurrence in CaM-binding proteins.

Figure 7.1 depicts the frequency of occurrence of SLiMs in the CaM-binding protein dataset. The figure shows that the SLiMs are present widely in the entire protein sequences with many binding regions overlapping with each other. The graph after the

100^{th} motif shows more clustered occurrences as the known motif family occurrence is much more present in the protein sequences. This is because the new motifs generated by MEME are independent of any existing motif family. These motifs generated are the binding regions for individual protein sequences and not something common across CaM-binding proteins.

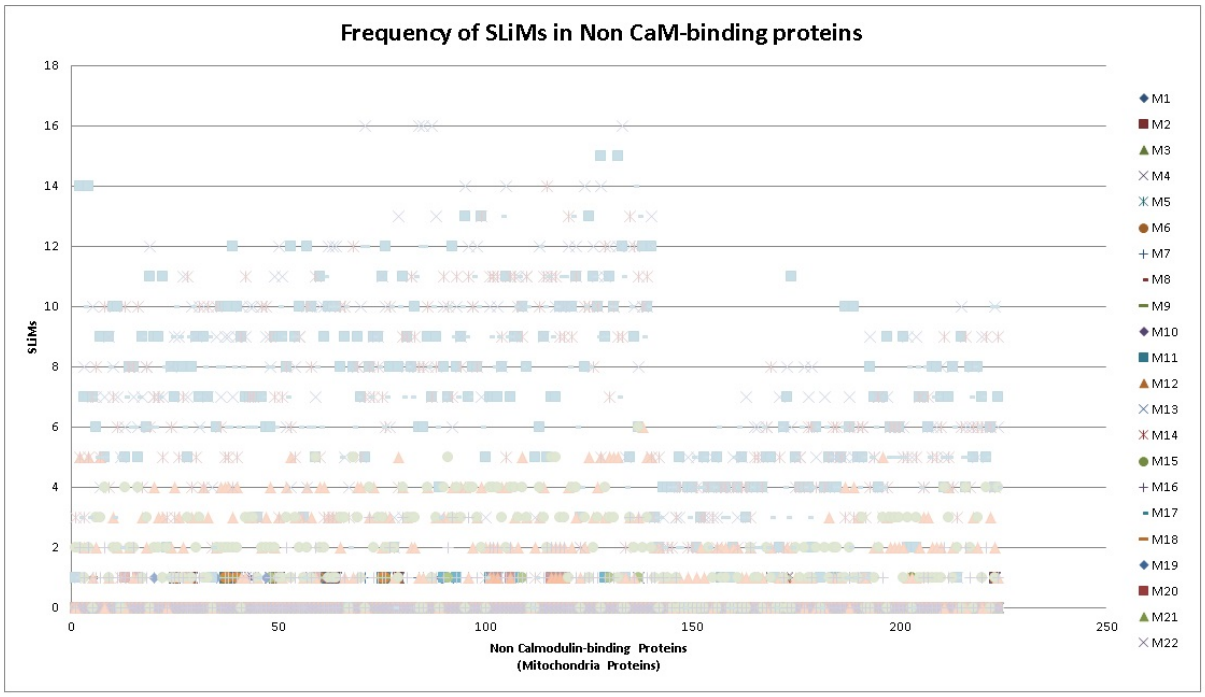

Figure 7.2: SLiM Frequency occurrence in Non CaM-binding proteins.

Figure 7.2 depicts the frequency occurrence of SLiMs in the Non CaM-binding proteins which are the Mitochondrial proteins. In this diagram, the new motifs or SLiMs discovered are scattered across all the protein sequences since their discovery is pertaining to the individual protein sequence. The clustered grouping is observed after the 100^{th} which clearly indicates the existence of known SLiMs across the protein sequences but less cluttered as compared to the CaM-binding proteins of Figure 7.1.

7.2 Experimental Results

The proposed model classifies the protein dataset into Calmodulin-binding and Non Calmodulin-binding using the SLiMs as the features for classification. The performance of the dataset is validated using 10-fold cross validation for SVM, k-NN and Naive Bayes classifiers.

The known and new SLiMs are used as the features for prediction of CaM-binding proteins. For k-NN, various values of k have been used for classification. The classification accuracies for k-NN for the dataset are shown in Table 7.2.

K-NN k =	Accuracy
1	94.49%
2	91.38%
3	94.25%
5	94.01%
7	94.01%

Table 7.2: k-NN classification accuracy for different values of k.

Table 7.2 represents the accuracy of k-NN using different values of k. The table shows that with higher values of k, the accuracy decreases and with $k = 1$ or $k = 3$ the accuracy rate is higher.

For SVM, I have used the kernel type as polynomial (degree = 3) and radial basis function and linear. The accuracy of classification is shown in Table 7.3.

SVM Kernel	Accuracy
Radial basis function	94.25%
Polynomial (degree = 3)	94.97%
Linear	94.97%

Table 7.3: SVM classification accuracy.

Table 7.3 depicts the accuracy for using the SVM classifier. It shows that as

the dataset is moved to higher dimension polynomially or linearly, a higher accuracy is achieved. Also changing the degree of the polynomial from 3 to 2 improves the accuracy from 94.25% to 94.97%.

For Naive Bayes, the batch size is changed from 100 to 200 with the Kernel Estimator value set to true. The accuracy of classification is shown in Table 7.4.

Parameters	Accuracy
Batch = 100	92.10%
Batch = 200	92.82%
KernelEstimator = true	92.82%

Table 7.4: Classification accuracy for Naive Bayes.

With the increase in batch size the accuracy increases combined with the KernelEstimator value set to true. This shows that the dataset when classified through batches achieves higher accuracy while using Naive Bayes classifier.

The ROC Curve as well as the Area Under the ROC curve (AUC) for each of the classifiers k-NN, SVM and Naive Bayes is shown in Figures 7.3, 7.4 and 7.5 along with their TP, FP, Precision, Recall and Specificity values.

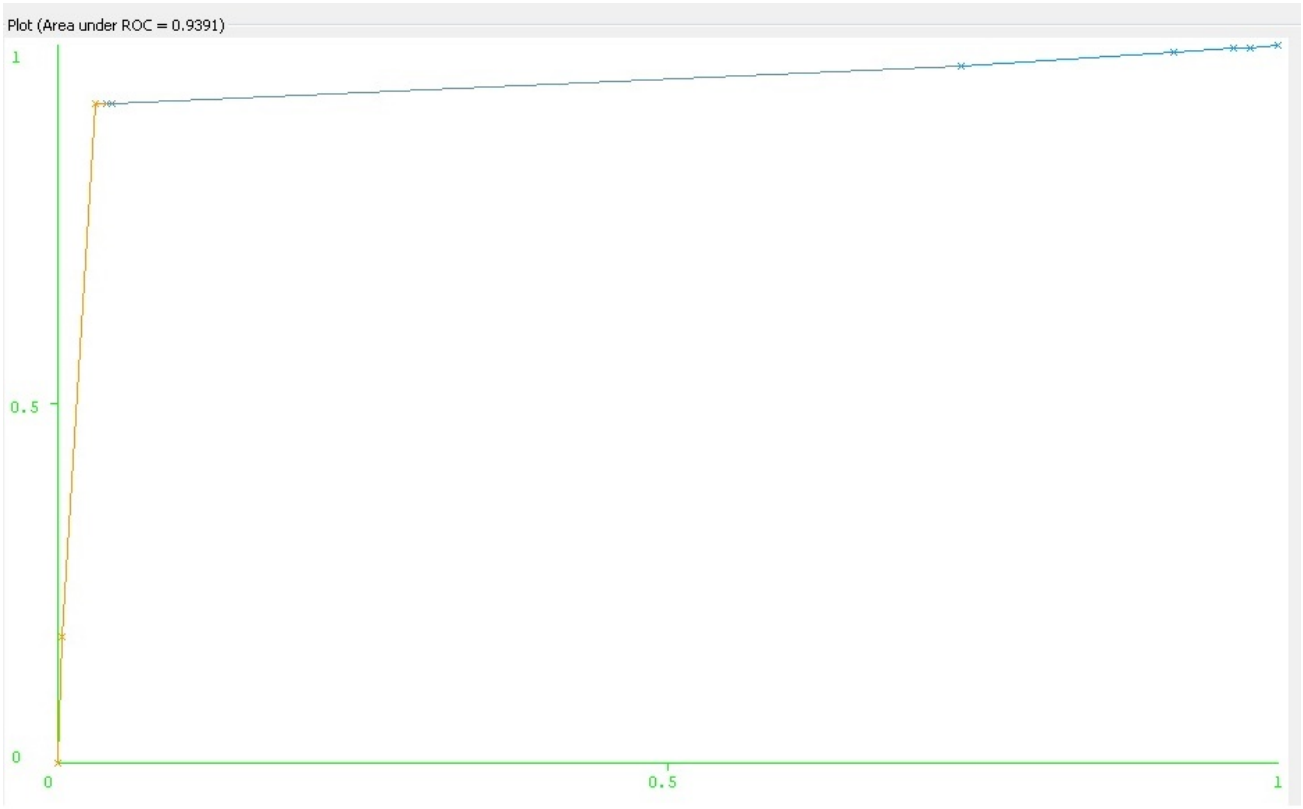

Figure 7.3: ROC curve for k-NN.

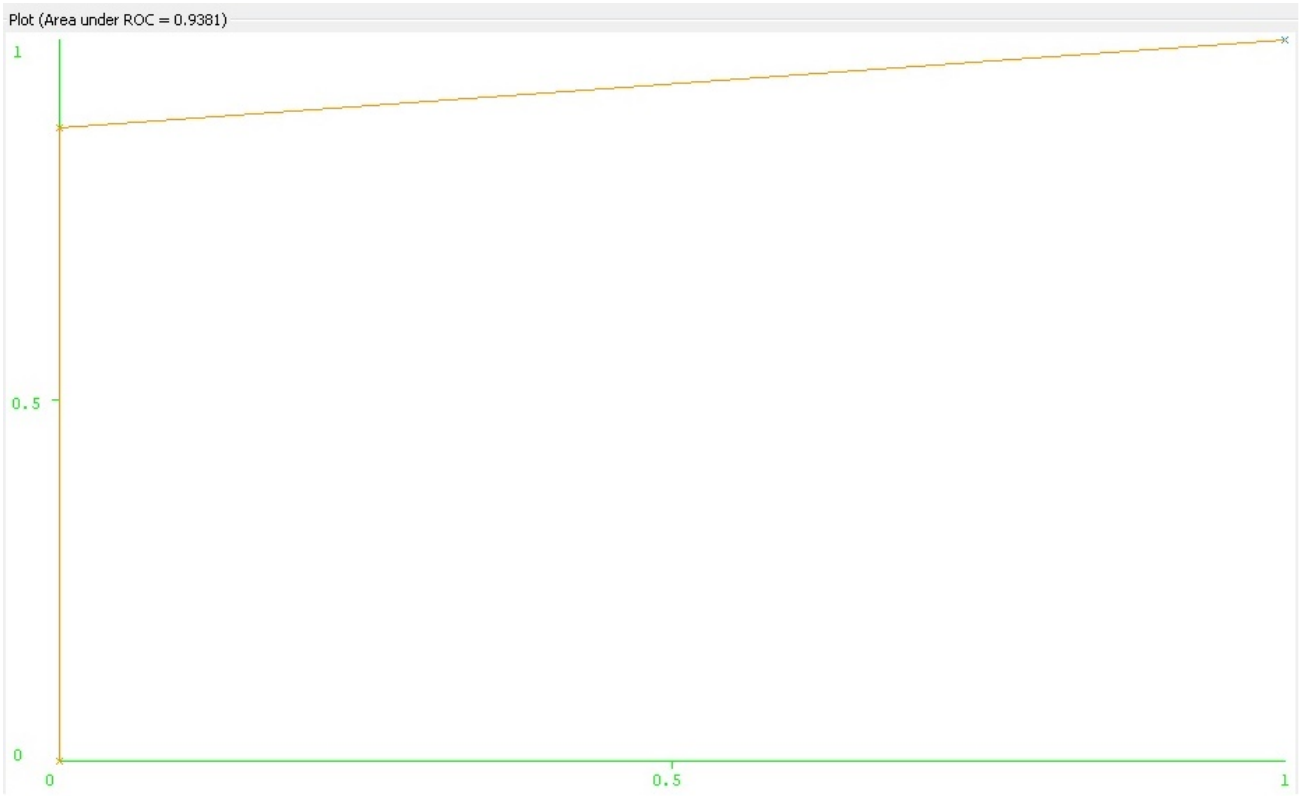

Figure 7.4: ROC curve for SVM.

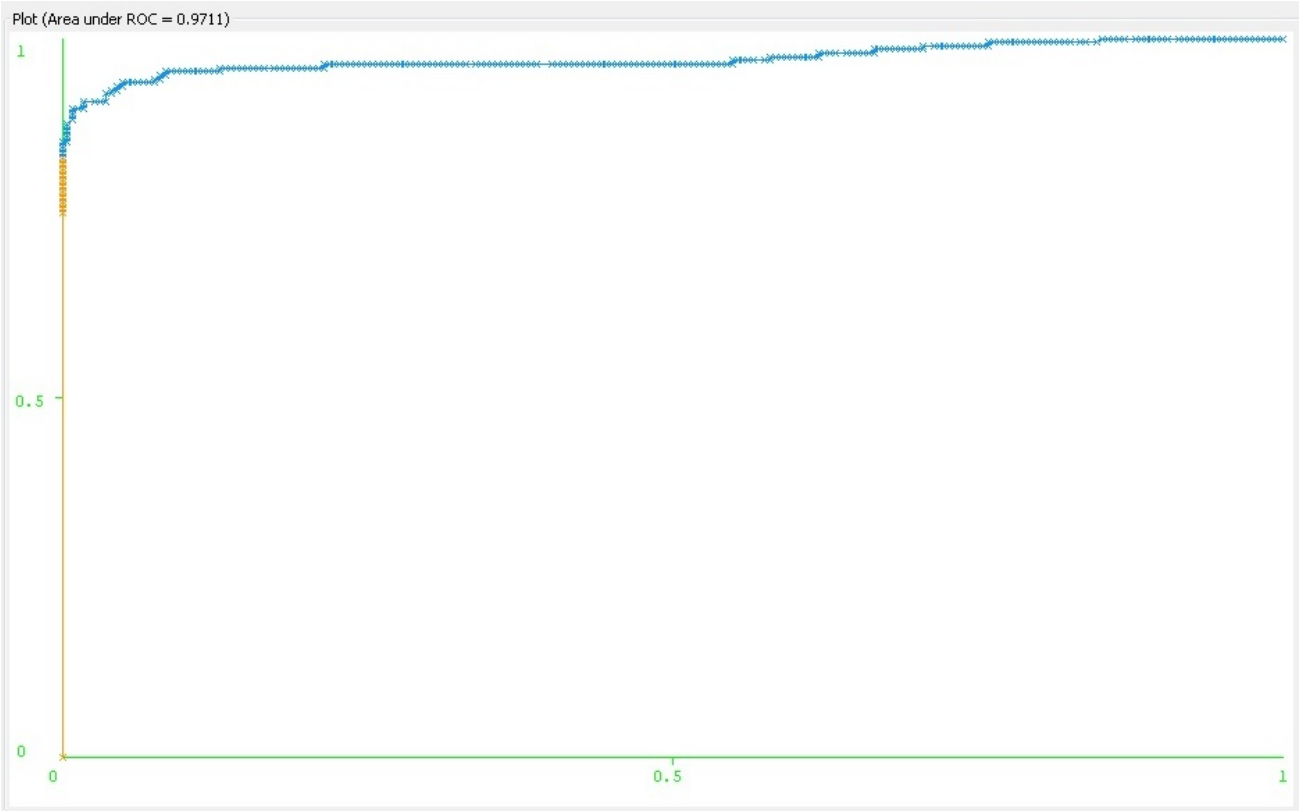

Figure 7.5: ROC curve for Naive Bayes.

Values	K-NN	SVM	Naive Bayes
TP	91%	89%	84%
FP	3%	0%	0%
PPV	96%	100%	100%
NPV	91%	89%	84%
Specificity	93%	94%	91%
Sensitivity	89%	90%	86%
Accuracy	94.49%	94.97%	92.82%

Table 7.5: Measures of performance for all classifiers.

The ROC curve for each of the classifiers have the curve towards the top-left corner which indicates very good classification results. The ROC curve towards the bottom indicate poor classification results. This trend of the curve shows that the classifier predicts the Calmodulin binding proteins using the SLiMs as their features for classification with best results. The performance achieved is higher and this proves that the proposed model is indeed suitable to be used for future biological analysis.

7.3 Discussion

Predicting Calmodulin-binding proteins through the Calmodulin-binding motifs has more importance in the computational field of molecular biology than predictions of CaM-binding sites. Due to this increase interest in Calmodulin and Calmodulin-binding proteins, the proposed model helps in quickly identifying the potential Calmodulin-binding proteins. The proposed model considers the overlapping of motifs present in the protein sequences which thereby helps identify CaM and CaM-binding proteins that bind to other peptide sequences.

Chapter 8

Conclusion

8.1 Summary of Contributions

In this thesis, I have introduced a prediction model to predict Calmodulin-binding proteins. I use Short Linear Motifs (SLiMs) to generate the features for classification. The important developments of the thesis are:

- The proposed model results in significant predictions of CaM-binding and Non CaM-binding proteins using SLiM patterns contained in the protein sequences as features for classification.
- This model denotes the Calmodulin binding regions in proteins and also finds the new concept of finding overlapping motifs. This feature of finding out the overlapping SLiMs and their regions in the protein sequences helps determine the classification of Calmodulin-binding proteins.
- Cross validation used in this thesis indicates the significance of accurately predicting the Calmodulin and non Calmodulin binding proteins.
- The analysis of SLiMs shows the binding regions in the protein sequences and also how new motifs can be generated for prediction of CaM-binding proteins. The analysis also highlights the distribution of SLiM sites in different protein sequences.

8.2 Limitations

This proposed model only takes into consideration the Calmodulin proteins in human organisms. Other species (plants, animals) are not taken into consideration. Also SLiMs are used to predict the Calmodulin-binding proteins. SLiMs are short patterns in protein sequences. The motifs of length of 10 or more amino acids are not included in this thesis. Thus Calmodulin-bindng proteins can be predicted for SLiMs of length 2 to 7. Also, this thesis does not include the scoring of motifs features which could also help in the classification and prediction of Calmodulin-binding proteins.

8.3 Future Work

I used only a few of the known SLiM families for the classification as their presence in the dataset of protein sequences were quite less. The remaining known SLiM families can be taken into consideration with more Calmodulin and non Calmodulin binding proteins apart from the proteins used in the dataset. I have also used SLiMs identified by MEME, whereas other SLiM identification tools are also available. Hence, other SLiM identification tools can be used to discover new motifs. While identifying SLiMs from the protein dataset, I used the length of 2 to 7 or 3 to 10 . Different parameters can be used with different other combinations to discover new motifs. Hence, the future work can be summarized as follows:

- Groups of SLiMs of different lengths, numbers and site number can be defined and used.
- Different combinations of other MEME parameters can be used to define different groups of SLiMs.
- Other SLiM identification tools can be used in order to discover new SLiMs.
- Longer motifs of different lengths can be used to predict the Calmodulin-binding proteins from the protein sequences.
- Other organisms' proteins can be used for the prediction of CaM-binding proteins apart from human proteins currently evaluated.
- Scoring functions can be used for the SLiMs for further analysis of CaM-binding proteins.

Appendix A

Discover SLiMs using MEME

Discovering new SLiMs using MEME is simple and unambiguous. MEME provides SLiM finding services online as a Web server as well as through installation of software packages in computer servers (Linux and Mac).

A.1 MEME Online Web Service

The online version of MEME is available at "*http://meme-suite.org/*". This version of MEME is quite user friendly. It has limitations in terms of defining various parameters as per the user requirements. It only allows the users to define the maximum and minimum width of a motif, maximum number of motifs and maximum and minimum number of sites. It does not support maximum number of motifs of more than 100 and does not accept a sequence dataset of more than 60,000 characters. It displays the output online and also has the facility to email at the email address provided by the user.

A.2 Installation of the Stand-alone Version

The installation version of MEME can be configured as per the user requirements. I configured it to support a maximum number of motifs = 100, and minimum width of a motif = 2. This is the default configuration of MEME.

MEME can be installed only on Linux and Mac Operating Systems with the prerequisite softwares already configured into the servers. The prerequisite software packages are:

- Perl (version 5.10.1 or higher)
  - Math::CDF
  - HTML::PullParser
  - HTML::Template
  - LWP
  - SOAP::Lite
  - XML::Simple
- Python (version 2.6 upto 2.7x)
- Zlib
- Bourne Shell or C Shell
- GNU compatible make
- C Compiler
- Gzip/gunzip
- ImageMagick
- Libexslt
- Libxml2
- Libxslt
- Libexpat1-dev

A.3 Input to MEME

The inputs to MEME are of two types: the first, is the entire input dataset and the second are the parameters to MEME.

A.3.1 FASTA Format

MEME takes FASTA format sequence files as input. The input file may contain the sequence for a single protein or a set of sequences of different proteins. A sample FASTA sequence file for protein complex Prolactin (P01236) is as follows:

```
>sp|P01236|PRL_HUMAN Prolactin OS=Homo sapiens GN=PRL PE=1
SV=1MNIKGSPWKGSLLLLLVSNLLLCQSVAPLPICPGGAARCQVTLRDLFDRAVVLSHYIHNLSSEMFSEFD
KRYTHGRGFITKAINSCHTSSLATPEDKEQAQQMNQKDFLSLIVSILRSWNEPLYHLVTEVRGMQEAPEAILS
KAVEIEEQTKRLLEGMELIVSQVHPETKENEIYPVWSGLPSLQMADEESRLSAYYNLLHCLRRDSHKIDNYLKL

LKCRIIHNNNC
```

Figure A.1: FASTA Sequence of Protein Prolactin P01236.

A.3.2 Parameters

MEME provides various parameters to meet the expected output. It takes one required parameter (i.e., the dataset parameter) and the others are all optional. MEME consists of many parameters though the parameters I used in this thesis are:

- dataset - file containing FASTA sequence format
- -o - name of directory for output files
- -protein - protein sequences
- -mod [oops, zoops, anr] specify any one

- -nmotifs nmotifs - maximum number of motifs to find
- -nsites nsites - number of motif sites
- -minsites minsites - minimum motif sites for each motif
- -maxsites maxsites - maximum motif site for each motif
- -w w - motif width
- -minw minw - minimum motif width
- -maxw maxw - maximum motif width

A.4 MEME Output

MEME provides three different output formats containing the motif information: HTML, XML and Text formats. The HTML file provides graphic or visualization formats compared to the other output files.

- **HTML**: All the motifs and its corresponding motif information are represented graphically. It contains the regular expression, starting motif location, ending motif location, all the sites and sequence logos of each motif. It also displays the block representation of all the sites.
- **TEXT**: It contains the complete information about each motif used in the experimentation. It contains regular expressions, start locations, end locations, and site information for a motif in the text format.
- **XML**: XML format file is useful mainly for parsing specific information for analysis. It shows regular expression, start position, end position, and site information of each motifs.

Bibliography

[1] Branden C, Tooze J. *Introduction to Protein Structure*. New York: Garland Pub. ISBN 0-8153-2305-0.

[2] S. Ren, G. Yang, Y. He, Y. Wang, Y. Li, & Z Chen. *The conservation pattern of short linear motifs is highly correlated with the function of interacting protein domains*. BMC genomics, 9(1)(2008), 1.

[3] Cino, Elio A., Wing-Yiu Choy, and Mikko Karttunen. *Conformational biases of linear motifs*. The Journal of Physical Chemistry B 117.50 (2013): 15943-15957.

[4] O'Day, Danton H. *CaMBOT: profiling and characterizing calmodulin-binding proteins*. Cellular signalling 15.4 (2003): 347-354.

[5] Norman E. Davey, Gilles Trave, Toby J. Gibson. *How viruses hijack cell regulation*. Trends in biochemical sciences, 2011, 36.3, 159-169

[6] Niall Haslam, Denis C. Shields. *Profile-based short linear protein motif discovery*. Bmc Bioinformatics, 2012.

[7] Stevens, Frits C. *Calmodulin: an introduction*. Canadian journal of biochemistry and cell biology 61.8 (1983): 906-910.

[8] Michael P. Walsh *Calmodulin and the regulation of smooth muscle contraction*. Molecular and cellular biochemistry 135.1 (1994): 21-41.

[9] Karen Mruk, Brian M. Farley, Alan W. Ritacco, William R. Kobertz. *Calmodulation meta-analysis: Predicting calmodulin binding via canonical motif clustering*. The Journal of general physiology, 2014, 144.1, 105-114.

[10] Bryan E. Finn, Sture Forsen. *The evolving model of calmodulin structure,function and activation*. Structure 3.1, 1995, 7-11.

[11] Manish Kumar Pandit *Prediction and analysis of protein-protein interaction types using short, linear motifs*. University of Windsor.

[12] Kyoko L. Yap, Justin Kim, kevin Truong, Marc Sherman, Tao Yuan, Mitsuhiko Ikura. *Calmodulin Target Database*. Journal of structural and functional genomics, 2000, 1.1, 8-14.

[13] T. Hunt *Protein sequence motifs involved in recognition and targeting: a new series*. Trends Biochem. Sci 15 (1990): 305.

[14] Ngoc Tam L Tran, Chun-Hsi Huang. *A survey of motif finding Web tools for detecting binding site motifs in ChIP-Seq data*. Biol Direct, 2014, 9.1.

[15] Timothy L. Bailey and Charles Elkan. *Fitting a mixture model by expectation maximization to discover motifs in biopolymers*. AAAI Press, Menlo Park, California, 1994, 28-36.

[16] Molsoft. *Protein Structure Tutorials*. 2016 `http://www.molsoft.com/gui/proteinstructure-tutorials.html`

[17] Costa V, Angelini C, De Feis I, Ciccodicola A. *Uncovering the complexity of transcriptomes with RNA-Seq*. J Biomed Biotechnol, 2010, 1-19.

[18] Zambelli F, Pesole G, Pavesi G. *Motif discovery and transcription factor binding sites before and after the next-generation sequencing era*. Brief Bioinform, 2012, 225-237.

[19] Zambelli F, Pesole G, Pavesi G. *PscanChIP: finding over-represented transcription factor-binding site motifs and their correlations in sequences from ChIP-Seq experiments*. Nucleic Acids Res., 2013.

[20] Timothy L. Bailey, N Williams, C Misleh, W Li. *MEME: discovering and analyzing DNA and protein sequence motifs*. Nucleic Acids Res., 2006, W369-W373.

[21] Frith M, Saunders N, Kobe B, Bailey T. *Discovering sequence motifs with arbitrary insertions and deletions*. PLoS Comput Biol., 2008.

[22] Sharov A, Ko M. *Exhaustive search for over-represented DNA sequence motifs with CisFinder*. DNA Res. 2009, 2009.

[23] Timothy L. Bailey *The MEME Suite* `http://meme-suite.org/index.html`.

[24] A. Mucherino, P. Papajorgji, and P. Pardalos. *k-nearest neighbor classification* Data Mining in Agriculture, 2009, 83-106.

[25] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, Inc, 2000, 2nd edition

[26] *Weka (Machine Learning)*
`https://en.wikipedia.org/wiki/Weka_ %28machine_learning%29`

[27] *Uniprot*. 2015.
`http://www.uniprot.org/`

[28] O'Day, Danton H. *CaMBOT: profiling and characterizing calmodulin-binding proteins*. Cellular signalling, 15.4, (2003), 347-354.

[29] Mooney, Catherine, Gianluca Pollastri, Denis C. Shields, and Niall J. Haslam. *Prediction of short linear protein binding regions*. Journal of molecular biology 415, no. 1 (2012): 193-204.

[30] Zambelli, Federico, Graziano Pesole, and Giulio Pavesi. *Motif discovery and transcription factor binding sites before and after the next-generation sequencing era*. Briefings in bioinformatics (2012): bbs016.

[31] M. R. Schiller, T. Mi, J. C. Merlin, S. Deverasetty, M. R. Gryk, T. J. Bill, ... & D. P Sargeant. *Minimotif Miner 3.0: database expansion and significantly improved reduction of false-positive predictions from consensus sequences*. Nucleic acids research (2011): gkr1189.

[32] H. Dinkel, K. Van Roey, S. Michael, M. Kumar, B. Uyar, B. Altenberg, ... & S. L Dahl. *ELM 2016data update and new functionality of the eukaryotic linear motif resource*. Nucleic acids research (2015): gkv1291.

[33] Jaime Davila, Sudha Balla, and Sanguthevar Rajasekaran. *Fast and practical algorithms for planted (l, d) motif search* Computational Biology and Bioinformatics, IEEE/ACM Transactions on 4.4 (2007): 544-552.

[34] C. J. Sigrist, E. De Castro, L. Cerutti, B. A. Cuche, N. Hulo, A. Bridge, ... & I Xenarios. *New and continuing developments at PROSITE*. Nucleic acids research (2012): gks1067.

[35] Norman E. Davey, Martha S. Cyert, and Alan M. Moses. *Short linear motifsex nihilo evolution of protein regulation* Cell Communication and Signaling 13.1 (2015): 1.

[36] N. E. Davey, N. J. Haslam, D. C. Shields, & R. J Edwards. *SLiMSearch 2.0: biological context for short linear motifs in proteins.* Nucleic acids research 39.suppl 2 (2011): W56-W60.

[37] N. E. Davey, N. J. Haslam, D. C. Shields, & R. J Edwards. *SLiMFinder: a web server to find novel, significantly over-represented, short protein motifs.* Nucleic acids research (2010): gkq440.

[38] Kevin T OBrien, Niall J. Haslam, and Denis C. Shields. *SLiMScape: a protein short linear motif analysis plugin for Cytoscape.* BMC bioinformatics 14.1 (2013): 1.

[39] N. E. Davey, K. Van Roey, R. J. Weatheritt, G. Toedt, B. Uyar, B. Altenberg, & T. J Gibson. *Attributes of short linear motifs.* Molecular BioSystems 8.1 (2012): 268-281.

[40] Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC. *Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment* Science. 1993, 262 (5131): 208-214. 10.1126/science.8211139.

Vita Auctoris

Mrinalini Pandit was born in 1989 in Mumbai, India. She completed her Bachelor's degree from the University of Mumbai, India in the field of Information and Technology in 2009. She later completed her Master's degree from the University of Mumbai in Information and Technology in 2012. Her area of interests in the field of research includes programming languages, machine learning, bioinformatics, prediction of Calmodulin and Calmodulin-binding proteins, discovery of Short Linear Motifs and pattern recognition.