2016

Prediction of High-throughput Protein-Protein Interactions and Calmodulin Binding Using Short Linear Motifs

Yixun Li
University of Windsor

Prediction of High-throughput Protein-Protein Interactions and Calmodulin Binding Using Short Linear Motifs

By

Yixun Li

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 High-throughput Protein-Protein Interactions and Calmodulin Binding Using Short Linear Motifs

by

Yixun Li

APPROVED BY:

S. Nkurunziza
Department of Mathematics and Statistics

A. Mukhopadhyay
School of Computer Science

A. Ngom, Advisor
School of Computer Science

L. Rueda, co-Advisor
School of Computer Science

Sep 15, 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 anyones 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

Prediction of protein-protein interactions (PPIs) is a difficult and important problem in biology. Although high-throughput technologies have made remarkable progress, the predictions are often inaccurate and include high rates of both false positives and false negatives. In addition, prediction of Calmodulin Binding Proteins (CaM-binding) is a problem that has been investigated deeply, though computational approaches for their prediction are not well developed. Short-linear motifs (SLiMs), on the other hand, are being effectively used as features for analyzing PPIs, though their properties have not been used in high-throughput interactions. We propose a new method for prediction of high-throughput PPIs and CaM-binding proteins based on counting SLiMs in protein sequences with specific scoring functions. The method has been tested on a positive dataset of 50 protein pairs obtained from the PrePPI database, and a negative dataset of 38 protein pairs obtained from the Negatome-PDB 2.0 database, and 387 proteins from the CaM database. We have used Multiple EM for Motif Elucidation (MEME) to obtain motifs for each of the positive and negative datasets. Our method shows promising results and demonstrates that information contained in SLiMs is highly relevant for accurate prediction of high-throughput PPIs and CaM-binding proteins. In addition to efficient prediction, individual SLiMs bring extra information on patterns that may be linked to specific roles in protein function.

AKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Dr. Alioune Ngom and Dr. Luis Rueda for their constant guidance and encouragement during my whole Master's period at the University of Windsor; without their valuable help, this thesis would not have been possible.

I would also like to express my appreciation to my thesis committee members Dr. Asish Mukhopadhyay, Dr. Sévérien Nkurunziza. Thank you all for your valuable guidance and suggestions to this thesis.

Meanwhile, I would like to thank Dr. Mina Maleki for all the help during my research process, including finding resources about Calmodulin Binding proteins and providing me very helpful guidance for the classification experiments.

Last but not least, I want to express my gratitude to my parents and my friends who give me consistent help over the past two years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Protein-protein Interaction

Comprehensive analysis of *protein-protein interactions* (*PPIs*) has been regarded as very significant for the understanding of underlying mechanisms involved in cellular processes [25]. PPIs are crucial for all biological processes [36]. While many proteins perform their functions when they interact with other proteins, understanding and studying PPIs is very important in almost all biological processes taking place in the cell, and help predict the function of unknown proteins [2].

PPIs networks provide a valuable framework for a better understanding of the functional organization of the proteome [36], and summarize large amounts of protein-protein interaction data, both from individual, small-scale experiments and from automated high-throughput screens [8]. Therefore, compiling PPI networks may provide new insights into protein function [36].

Common high-throughput experimental techniques for predicting PPIs such as *Yeast two-hybrid* (*Y2H*) [44] and *Tandem Affinity Purification* (*TAP*) [19] have enabled the production of large amounts of PPI data [20]. Nevertheless, these techniques are expensive, labor-intensive, suffering from insufficient coverage [45] and usually lead to high false-positive and false-negative rates [2]. Thus, developing reliable computational approaches to predict PPIs is of great significance.

1.2 Calmodulin Binding Proteins

Calmodulin (CaM) is a calcium-binding protein that is a major transducer of calcium signaling [37]. It has no enzymatic activity on its own but rather acts by binding to and altering the activity on a panel of cellular protein targets. Its targets are structurally and functionally diverse and participate in a wide range of physiological functions including immune response, muscle contraction and memory formation.

Figure 1.2.1 is a typical of calcium-dependent protein interaction, where the two halves of CaM bind to opposite sides of the target peptide (the four calcium molecules are green spheres). Identifying CaM target proteins and CaM sites is an important and ongoing research problem because of the great diversity of conformations it uses in its target interactions. This diversity cannot be captured by a single amino acid sequence motif, but instead CaM-binding sites are commonly divided into four or more motif classes with different sequence characteristics [43]. Current algorithms struggle to identify novel CaM-binding proteins.

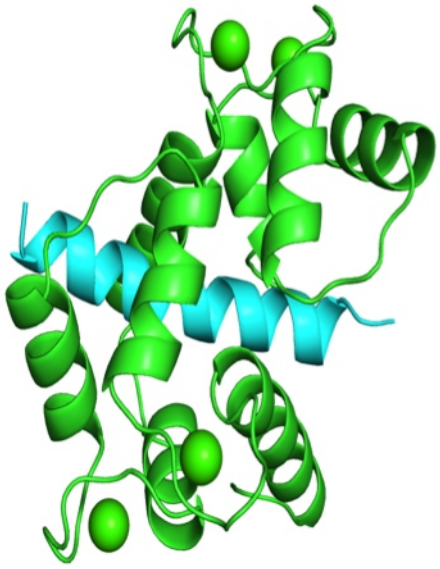

FIGURE 1.2.1: Structure of CaM (green) interacting with its binding domain from calcineurin (blue).

1.3 Motifs

A motif is a sequence pattern of nucleotides in a DNA sequence or amino acids in a protein [42]. Motifs are patterns widespread over a group of proteins that are related by function or may have other biological features in common. Given a set of sequences, motifs are common subsequences, which appear the most among these sequences. Usually, each motif contains a sequence pattern of 3-20 amino acids [29].

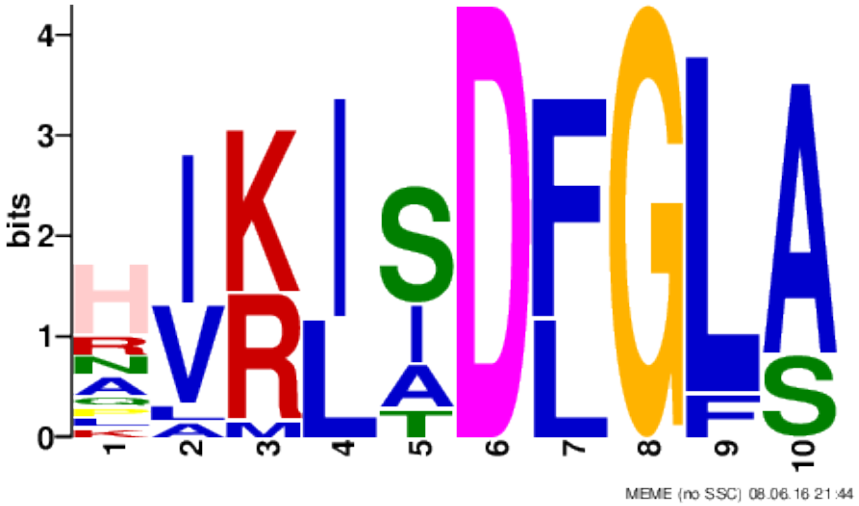

FIGURE 1.3.1: An amino-acids motif pattern.

1.3.1 Short Linear Motifs

Short-linear sequence motifs (*SLiMs*) or minimotifs in protein sequences are short patterns of 3-10 amino acids that have been found to be interesting [5], because of their capacity to encode functional aspects, bind to important domains and enrichment in intrinsically disordered regions of protein sequences [28]. They help regulate many cellular processes, by being interaction sites for other SLiMs containing proteins. SLiM-mediated interactions are often transient interactions or utilize additional interaction domains to co-operatively produce stable complexes. Therefore, prediction and analysis of PPIs and CaM-binding proteins using SLiM profiles has the potential to develop better models for cellular processes such as modulation and regulation of proliferation and apoptosis [18].

1.3.2 Tools for Finding Motifs

Motifs may be indistinguishable from random artifacts, therefore, discovering biological motifs in a set of sequences is a difficult task [6], and hence several approaches have been proposed for improving motif discovery [6], such as using *auxiliary data*, *PSP approach* and *Gibbs Sampling*.

From different accessible SLiM discovering tools such as SLiMFinder [12], SLiMSearch [13], Minimotif Miner (MnM) [33], and MEME (Multiple EM for Motif Elucidation) [4], MEME can discover SLiMs through an unsupervised approach and 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. It discovers motifs by optimizing the statistical parameters of the model using the Expectation Maximization (EM) algorithm, and a statistical sequence model to determine the positions and the width of the motif sites in the sequences [6].

MEME provides both a Web server online and a stand-alone application that can be dowloaded and installed locally on Unix or Linux platforms. The MEME Suite Web server provides a unified portal for online discovery and analysis of sequence motifs representing features such as DNA binding sites and protein interaction domains. The popular MEME motif discovery algorithm is now complemented by the GLAM2 algorithm which allows discovery of motifs containing gaps. Three sequence scanning algorithms MAST, FIMO and GLAM2SCAN allow scanning numerous DNA and protein sequence databases for motifs discovered by MEME and GLAM2 [3]. Figure 1.3.2 shows the Motif-based sequence analysis tools in the MEME Suite Website.

For this thesis, we downloaded MEME version 4.10.1 and installed it on Unix platform. First we obtained the protein sequences and placed them in a FASTA format [38] file in order to make it as an input file for MEME. In the shell script command line, we can set the input file name, output folder name, the number of the motifs and the length range of the motifs.

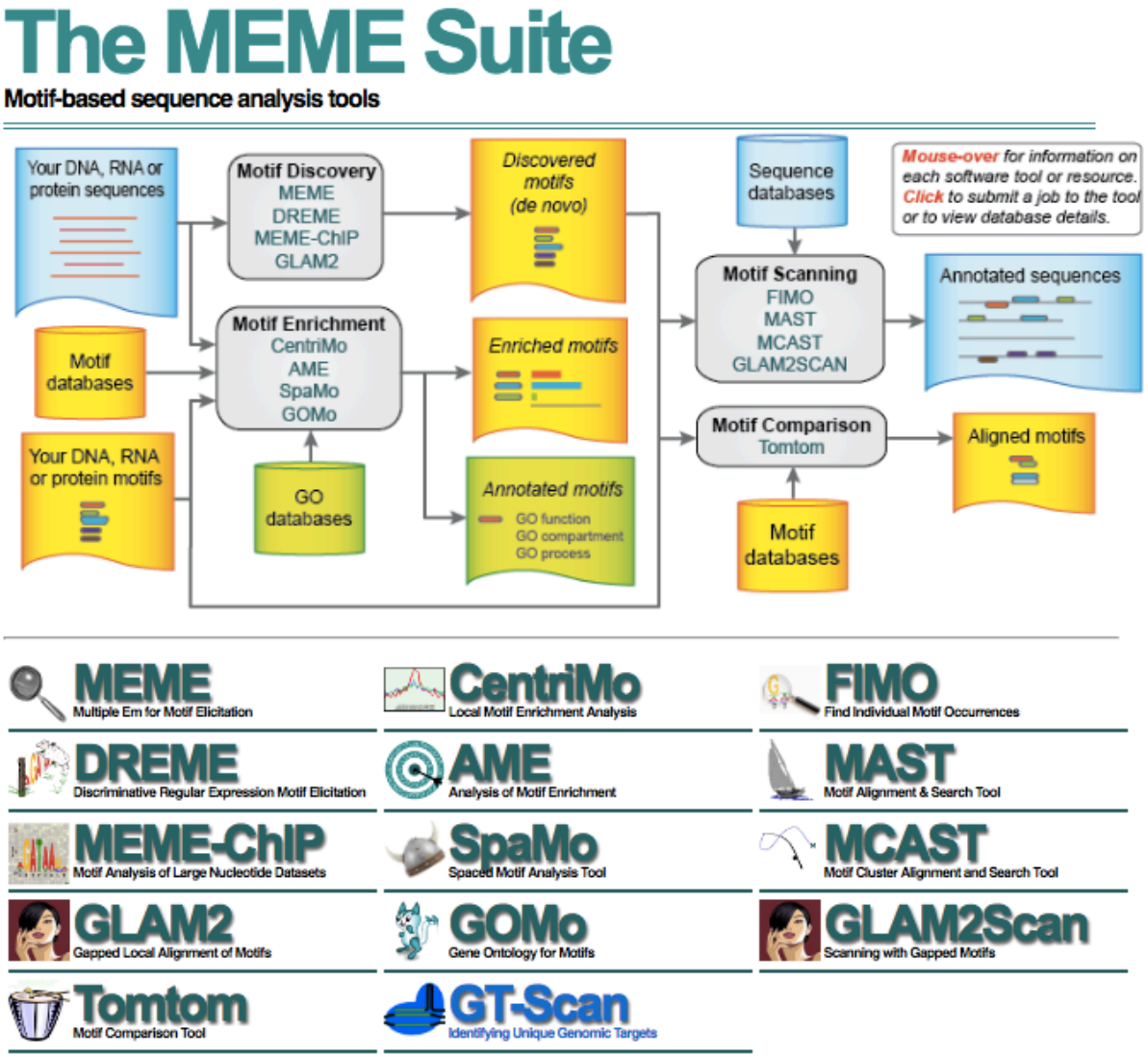

FIGURE 1.3.2: The MEME Suite. Figure obtained from meme-suite.org.

1.4 Tools for score processing

After we obtained the SLiMs from the protein datasets, we applied score processing for obtaining the score matrices for experiments. For processing the scores and the matrices, we use Python and Matlab for programming and matrix operations.

1.4.1 Python

Python is a popular programming language for scientific computing. It provide state-of-the-art implementations of many well known machine learning algorithms, and maintains an easy-to-use interface. Therefore, the need grows for statistical data analysis by non-specialists in the software and Web industries, as well as in fields outside of computer-science, such as biology or physics [26].

There are plenty of data analysis libraries and tools for Computational Biology written in Python, which can be downloaded for free from http://www.biopython.org, such as Biopython [11]. Biopython includes modules for reading and writing different sequence file formats and multiple sequence alignments, dealing with 3D macro molecular structures, interacting with common tools such as BLAST, ClustalW and EMBOSS, accessing key online databases, as well as providing numerical methods for statistical learning [11]

For this thesis, we chose Python for programming, because it has an XML library ElementTree, which is very convenient for parsing XML trees. We downloaded the XML files for the proteins in the datasets from UniProt, which is a freely accessible database of protein sequence and functional information, then used Python XML parser to obtain the protein sequences. After we obtained the SLiMs of the proteins, we use Python regular expression to find the sites in the protein sequences (The sites for corresponding proteins are output by MEME, then we use regular expression to find sites in other protein sequences which in other datasets). At the end, we also use Python to process the scores and create the output matrices.

1.4.2 Matlab

Matlab is a data analysis and visualization tool that has been designed with support for matrices and matrix operations. Matlab has excellent graphics capabilities, and its own powerful programming language. One of the reasons that Matlab has become such an important tool is through the use of sets of Matlab programs designed to support a particular task [22].

In this thesis, we use Matlab to devide the elements in one matrix by the elements in another matrix, where that elements are in the same position in their own matrix.

1.5 Machine Learning

Learning processes include the acquisition of new declarative knowledge, the development of motor and cognitive skills through instruction or practice, the organization of new knowledge into general, effective representations, and the discovery of new facts and theories through observation and experimentation [9]. As one of the most challenging goal in artificial intelligence, researchers have been striving to implant such capabilities in computers and make the machines learn new knowledge. This field is called Machine Leaning (ML).

1.5.1 Tool for classification: WEKA

Waikato Environment for Knowledge Analysis (Weka) is a collection of ML algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from other Java programs [17].

Weka provides a general-purpose environment for automatic classification, regression, clustering and feature selection and common data mining problems in bioinformatics research. It contains an extensive collection of machine learning algorithms and data preprocessing methods complemented by graphical user interfaces for data exploration and the experimental comparison of different machine learning techniques on the same problem. Weka can process data given in the form of a single relational table. Its main objectives are

to (a) assist users in extracting useful information from data and (b) enable them to easily identify a suitable algorithm for generating an accurate predictive model from it [14].

Weka is a flightless bird with an inquisitive nature, it is found only on the islands of New Zealand. The name is pronounced like this, and the bird sounds like the one shown in Figure 1.5.1, which shows the logo of the Weka software.

FIGURE 1.5.1: WEKA software logo.

1.5.2 Classification algorithms

Weka provides many classification methods such as BayesNet, NaiveBayes, LibSVM with linear/polynomial/radial basis function (RBF) kernel, RBFNetwork, Multilayer Perceptron, k-Nearest Neighbor (kNN), Random Forest and Decision Tree, etc. In this thesis, we used different classifiers: LibSVM + Polynomial, LibSVM + RBF, Random Forest, kNN, Decision Tree and Multilayer Perceptron.

LibSVM is a library for Support Vector Machines (SVM) [10]. It is a powerful, state-of the art algorithm that can guarantee the lowest true error due to increasing the generalization capabilities [34]. LibSVM + linear was considered for our classification, as shown in Figure 1.5.2. Here, there are many possible linear classifiers that can separate the data, but there is only one that maximizes the margin (maximizes the distance between it and the nearest data point of each class). This linear classifier is termed the optimal separating hyperplane. Intuitively, we would expect this boundary to generalise well as opposed to other possible boundaries [15].

However, since the datasets are non-linear models, it is better to choose SVM + Polynomial or RBF kernel. The Polynomial kernel represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of

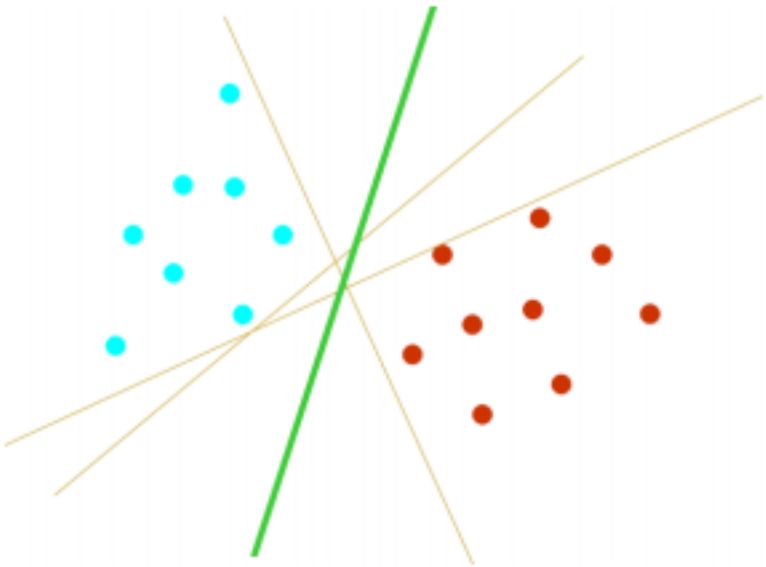

FIGURE 1.5.2: Optimal Separating Hyperplane [15].

non-linear models [41]. The RBF kernel is also commomly used in classifing non-linear models.

In SVM + Polynomial or RBF kernel, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. The C parameter trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly by giving the model freedom to select more samples as support vectors. In order to find the best parameters when using SVM classifier, we implied grid search with different C and gamma.

Random Forest (RF) is a classifier that is based on a combination of many decision tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [34]. RF has excellent accuracy among current classification algorithms. It also has an effective method for estimating missing data and maintains accuracy when a large proportion of the data is missing [34].

K-nearest-neighbor (kNN) is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. kNN was developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine [27].

Decision tree (DT) classifier is one of the possible approaches to multistage decision making. It is a way of representing a series of rules that lead to a class or value [34]. The basic idea involved in any multistage approach is to break up a complex decision into a union of several simpler decisions, hoping that the final solution obtained in this way would resemble the intended desired solution [31].

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training the network. MLP is a modification of the standard linear perceptron and can distinguish data that are not linearly separable [40].

1.5.3 Feature selection

During the last decade, the motivation for applying feature selection (FS) techniques in bioinformatics has shifted from being an illustrative example to becoming a real prerequisite for model building. In particular, the high dimensional nature of many modelling tasks in bioinformatics, going from sequence analysis over microarray analysis to spectral analyses and literature mining has given rise to a wealth of FS techniques being presented in the field [30].

Wrapper methods embed the model hypothesis search within the feature subset search. In this setup, a search procedure in the space of possible feature subsets is defined, and various subsets of features are generated and evaluated. The evaluation of a specific subset of features is obtained by training and testing a specific classification model, rendering this

approach tailored to a specific classification algorithm [30]. In this thesis, we applied the wrapper approach with RF for FS followed by classification using different algorithms.

1.5.4 Evaluation method

The K-fold cross validation refers to testing procedure where the dataset is randomly divided into K disjoint blocks of objects. Then the data mining algorithm is trained using k - 1 blocks and the remaining blocks is used to test the performance of the algorithm. This process is repeated k times. At the end, the recorded measures are averaged. It is common to choose k=10 or any other size depending on the size of the original dataset [34]. In this thesis, since the datasets are all not very large, we chose k=3 for evaluation.

We used the following performance measures: *Accuracy*, *Recall* and *Matthews correlation coefficient* (*MCC*) in order to assess the predictive capability of our approach, because the the accuracy of random classifiers is 50% for balanced distributions [32] and a coefficient of +1 represents a perfect prediction, 0 no better than random prediction and 1 indicates total disagreement between prediction and observation [39].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1.5.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{1.5.2}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \tag{1.5.3}$$

where TP is the number of True Positives, FP is the number of False Positives; TN is the number of True Negatives, and FN is the number of False Negatives. CaM-binding proteins are positive while Mitochondrial proteins are negative.

1.6 Motivation of this Thesis

Prediction of protein-protein interaction (PPI) is a difficult and important problem in biology. Although high throughput technologies have made remarkable progress, the predictions are often inaccurate and include high rates of both false positives and false negatives. Meanwhile, prediction of Calmodulin-binding (CaM-binding) proteins plays a very important role in the fields of biology and biochemistry, because Calmodulin binds and regulates a multitude of protein targets affecting different cellular processes.

Short-linear motifs (SLiMs) in protein sequences have being effectively used as features for predicting and analyzing obligate protein interactions, several computational approaches have been used for prediction of high-throughput PPIs, though their properties have not been used in the prediction of CaM-binding proteins, and none of them has exploited the power of SLiMs. In this thesis, we propose five new methods for prediction of PPIs and CaM-binding proteins based on counting scores of SLiMs between pairs of protein sequences with specific scoring functions.

The extracted features are new SLiMs derived from MEME. Two different approaches have been used to discover new motifs using MEME: (a) find SLiMs from each of the positive and negative datasets separately (SM) and (b) find SLiMs from the combined positive and negative datasets (CM).

As for prediction of PPIs, given two initial datasets of PPI pairs and non-PPI pairs, we first pre-processed the datasets into new smaller datasets as 50 PPI pairs and 38 non-PPI pairs for obtaining the SLiMs using MEME. We have used MEME to obtain 50 motifs for each of the positive and negative datasets, separately, obtaining a set of 100 motifs (the SM approach). Similarly, we generated 50 motifs from the combined negative and positive datasets (the CM approach).

For prediction of CaM-binding proteins, the dataset has been manually curated with 194 CaM-binding proteins as a positive dataset and 193 Mitochondrial proteins as a negative dataset. We have used MEME to obtain 50 motifs for each of the positive and negative datasets, separately, obtaining a set of 100 motifs (the SM approach). Similarly, we generated 100 motifs from the combined negative and positive datasets (the CM approach).

Predictions of CaM-binding proteins have been performed in the Waikato Environment for Knowledge Analysis (WEKA) using k nearest neighbor (k-NN), Support support vector machine (SVM), random forest (RF), decision tree (DT) and Multilayer Perceptron (MP) classifiers, on a 3-fold cross-validation setup. Moreover, the wrapper criterion with Random Forest for feature selection (FS) has been applied followed by classification using different algorithms mentioned above.

Our method shows itself to be very promising and demonstrates that information contained in SLiMs is highly relevant for accurate prediction of PPIs and CaM-binding proteins. In addition to efficient prediction, individual SLiMs may bring extra information on meaningful patterns linked to specific roles in protein function.

In the following chapters, we discuss about related works of prediction of PPIs and motifs in chapter 2, and we describe the datasets and method used in this thesis in Chapter 3. After that, we show the classification results and the analysis in Chapter 4, and we concludes the whole work in chapter 5.

CHAPTER 2

Review of the Literature

Recent studies have focused on the approaches of prediction of PPIs, the discovery of new SLiMs, and the prediction of protein interactions using SLiMs. In this chapter, we review the previous research and publications on prediction of Protein-protein interactions and research on Short Linear Motifs.

2.1 Approaches for Prediction of PPIs

In this section, we review two papers related to prediction of protein interactions using protein sequence informations. The first paper proposes a codon pair usage-based PPI prediction method. The second paper proposes a new method based on animo acid defferences between pairs of protein sequences.

2.1.1 Prediction of PPIs using information from simple condon pairs

The authors of [45] analyze the relatiionship between codon pair usage and PPIs in yeast, and show that codon pair usage of interacting protein pairs deffers significantly from randomly expected. This motivates the development of a novel approach for predicting PPIs, CCPPI, with codon pair frequency defference as input to a SVM classifier. The results show that CCPPI performs better than other sequence-based encoding schemes.

Previous work and shortcomings by others referred to by the authors

The authors state that it have been revealed so far that using high-throughput experimental techniques like yeast two-hybrid screening and tandem-affinity purification coupled with

mass spectrometry, miniatures of the interactomes of a few model organisms. However, the authors note that the experimental methods mentioned above are relatively expensive and labor intensive and suffering from insufficient coverage.

The new idea that the authors proposed

The authors proposed a codon pair usage-based PPI prediction method termed as CCPPI (Codon Combination-based Protein-Protein Interaction predictor) under the Support Vector Machine (SVM) framework. Their analyses show that codon pair usage of interacting protein pairs is significantly different from that of random protein pairs.

Materials and methods

They downloaded protein sequences and the corresponding coding sequences of yeast from the SGD database. They used three kinds of combined datasets of 4156 DIP positives and equal number of non-interacting protein pairs. The first kind of datsets that contains randomly selected non-interacting protein pairs as negatives are termed as "DIP + Random". The second kind ("DIP + RSS Negative") contains "RSS Negative" without known similar functions or subcellular localizations. The "RSS Negative" datasets were randomly selected protein pairs whose RSS (Biological Process) and RSS (Cellular Component) were less than 0.4. With respect to the third kind of datasets ("DIP + Homogeneous"), the negatives were generated by randomly rewiring the DIP positives.

They calculated the difference in a feature between a pair of proteins in specific scoring functions. They compared two previously published encoding schemes with their encoding schemes, CT encoding and AC encoding. They used the two encoding schemes to concatenate feature vectors for a pair of proteins instead of calculating the differences between them.

SVM predictors trained with codon pair frequency differences and other encodings were tested by 10-fold cross-validation using the three kinds of combined datasets which mentioned above. All SVM models were constructed with the RBF kernel using the LIBSVM package. The parameter C was preliminarily optimized to 10 and the other SVM parameters were set to their default values. All the the three encoding schemes (CCPPI,

CT encoding and AC encoding) perform better with C $= 10$ than the default C. They use the following four perfomance measures for evaluating the results: accracy, precision, sensitivity and MCC. The definition of precision and sensitivity are:

$$precision = \frac{TP}{TP + FP} \tag{2.1.1}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{2.1.2}$$

Results and discussion

The authors claim to have compared codon pair frequency differences between 4,380 interacting protein pairs from the DIP database and randomly selected protein pairs, which are 19-fold larger than the former. In total, there are $61 \times 61 = 3721$ codon pairs under investigation. Compared with randomly selected protein pairs, 1551 out of 3721 codon pairs in the interacting protein pairs were observed to have significantly similar frequencies. At the same significance level, the frequencies of 619 codon pairs in interacting protein pairs tend to be dissimilar. Moreover, there is a considerable fraction (41.7%) of codon pairs that do not have any significant difference. In contrast, 57 out of 61 codons in the interacting protein pairs show similar frequencies, which is consistent with previous observations based on a different dataset. They also claim that a predictor based on codon pair frequency differences may perform better in distinguishing interacting protein pairs from random protein pairs.

They also compared the performance of CCPPI and the other sequence-based encoding schemes through 10-fold cross-validation tests under the same SVM framewor, for a fair comparison of the performance of CCPPI and other different encoding schemes. The comparison results show that the codon frequency difference encoding outperformed the amino acid frequency difference encoding. Besides, CCPPI achieved a better performance than the amino acid pair frequency difference encoding. Moreover, they also compared the performance of CCPPI with other encoding schemes: CT encoding and AC encoding, and the accuracies for these two encodings are about 5-10% lower compared with CCPPI.

Their results indicate that the developed codon pair based method CCPPI is capable of predicting protein-protein interactions, with a favorable or at least competitive performance in comparing with several well-known sequence-based encoding schemes.

2.1.2 Prediction of PPIs using information from protein sequences

[21] describes our previous research related to prediction of PPIs based on amino acid differences between pairs of protein sequences. Our finding suggests that amino acid differences of interacting protein pairs are relevant to the prediction of PPIs and hence provide important imformation on sequence-based encoding schemes.

Previous work and shortcomings by others referred to by the authors

We state that the methods only using the information of protein sequences are more universal than those that depend on some additional information or predictions about the proteins. Paper [16] achieved acceptable performance on the yeast dataset using AC encoding of physicochemical features derived from spaced amino acid pairs. Paper [35] proposed a CT encoding scheme based on the calculation of tri-peptide frequencies, and it was shown to achieve good results in the human PPI dataset. However, though several sequence-based methods have shown that the information of amino acid sequences alone may be sufficient to identify novel PPIs, the highest prediction accuracy of these methods is only 80%. The information of the interactions which occurs in the discontinuous amino acids segments in the sequence may be able to further improve the prediction ability of the existing sequence-based methods.

Materials and methods

The positive reference set used in our dataset is obtained from the PrePPI (structure-based prediction of protein-protein interactions) database, from which we downloaded the New Human Protein Interaction Set and then randomly selected 4,000 positive pairs from the set. The negative reference set is obtained from the Negatome Database version 2.0, which is a repository of non-interacting pairs of proteins, and then we also randomly selected

4,000 negative pairs from the Protein Data Bank (PDB) in the Negatome Database. We downloaded the protein sequences from Uniprot.

After obtaining the positive and negative protein sequences datasets we calculated the difference of the counting of different amino acids between each positive pair and each negative pair of proteins, and used the difference of each amino acid as the features for each pair of proteins.

Experiments and analysis

We applied Naive Bayes, kNN, DT, RF and SVM with different kernels (Linear, Polynomial and RBF) classifiers on our datasets using WEKA. 10-fold cross-validation is the method we used for validating all the classifiers. We used the following performance measures in order to assess the predictive capability of our approach: accuracy, recall, FP rate, precision, F-measure and MCC.

We used the wrapper approach with mRmR (Minimum Redundancy and Maximum Relevance) which is available in WEKA. Random Forest was used within this wrapper for evaluating the accuracy of a feature subset.

Results that the authors claim to have achieved

The accuracy results of the classifiers on the original full dataset and on the reduced dataset (after FS) are, respectively, 87.2% and 86.3% for kNN, 89.3% and 89.4% for Random Forest, and 91.7% and 90.3% for SVM-RBF (see Figures 2.1.1, 2.1.2, 2.1.3, and 2.1.4). The ROC curve for SVM-RBF is shown in Figures 2.1.5.

We have shown that this simple encoding of protein pairs as difference vectors of amino-acid frequencies between protein pairs yield excellent results when using kNN, RF or SVM-RBF classifiers. Our results also show that FS is not necessary with this simple encoding. After these experiments, we considered to investigate complex but more discriminative encoding of protein pairs, such as counting the differences of multiple-gram amino acids rather than just counting the difference of 1-gram amino acid between pairs of proteins. Thus, we considered using SLiMs instead of 1-gram amino acid in our future works.

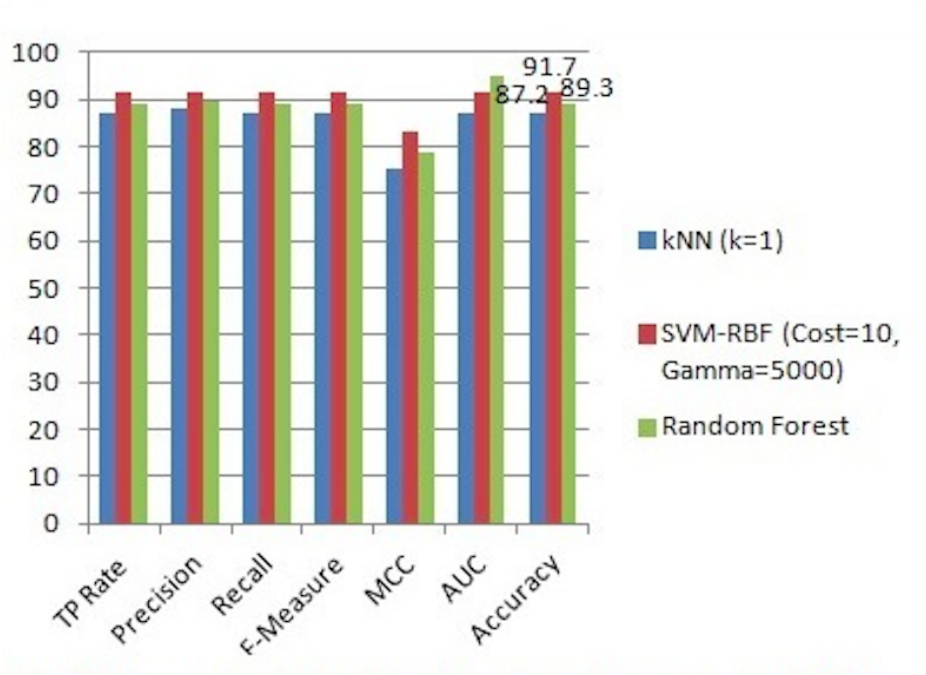

FIGURE 2.1.1: The results of classifying the original dataset using 1NN, SVM-RBF (Cost = 10 and Gamma = 5000), and Random Forest classifiers.

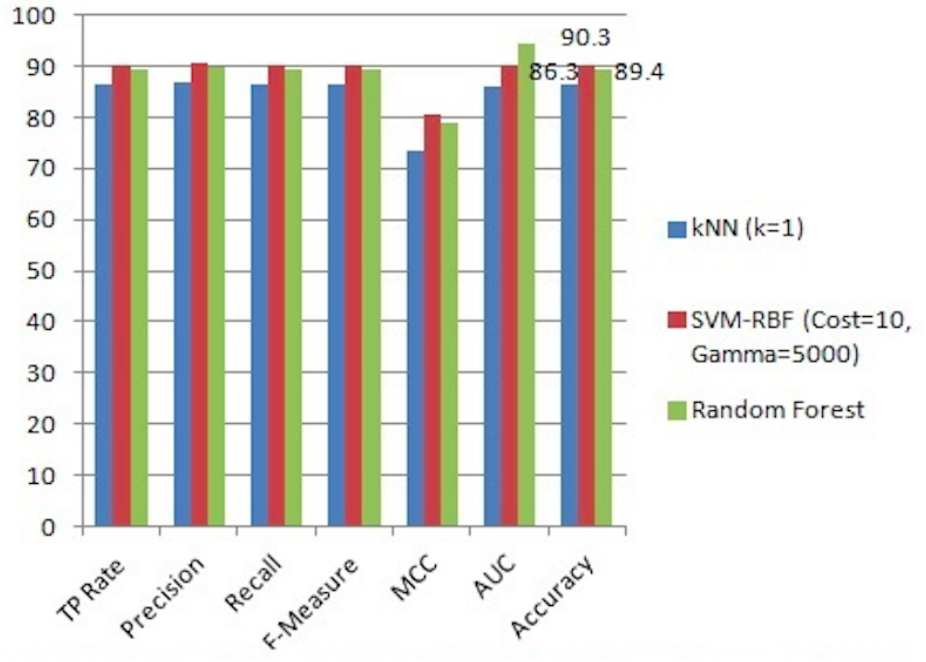

FIGURE 2.1.2: Using 1NN, SVM-RBF (Cost = 10 and Gamma = 5000), and Random Forest to classify the dataset obtained after applying feature selection.

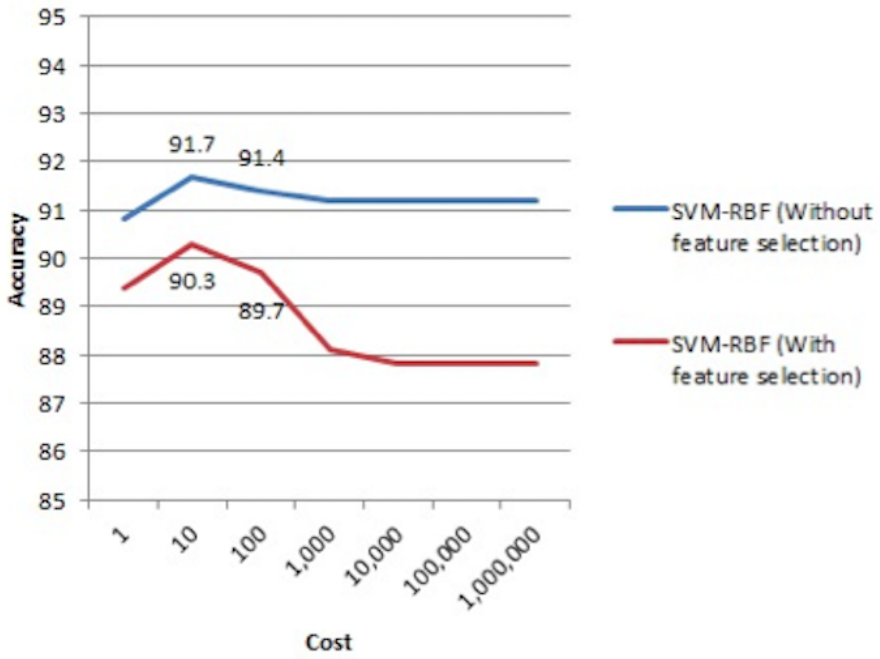

FIGURE 2.1.3: Results of using SVM-RBF classifier (with Gamma fixed to 5000) based on accuracy on both original and selected feature datasets.

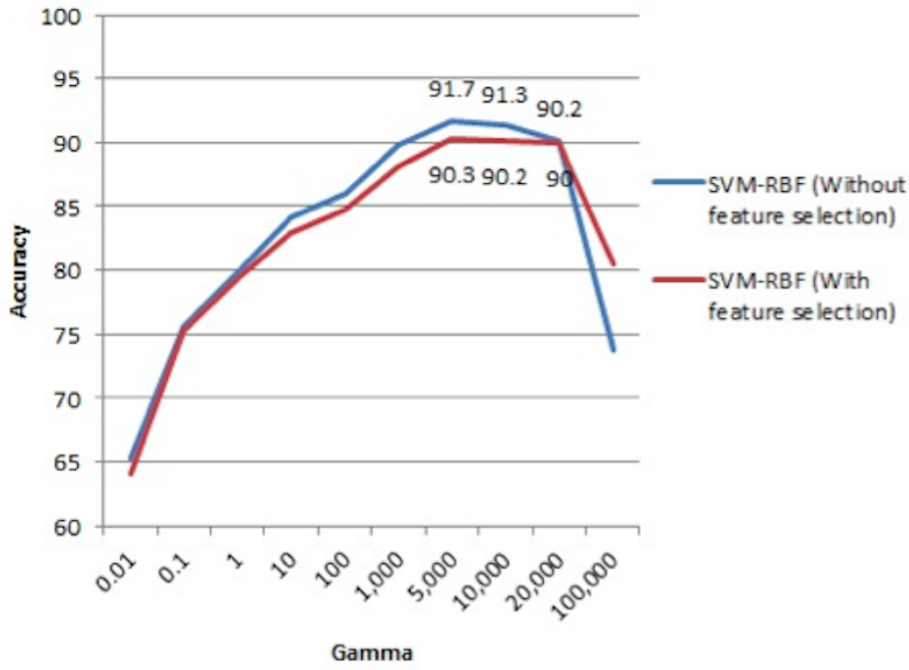

FIGURE 2.1.4: Results of using SVM-RBF classifier (Cost fixed to 10) based on accuracy on both original and selected feature detasets.

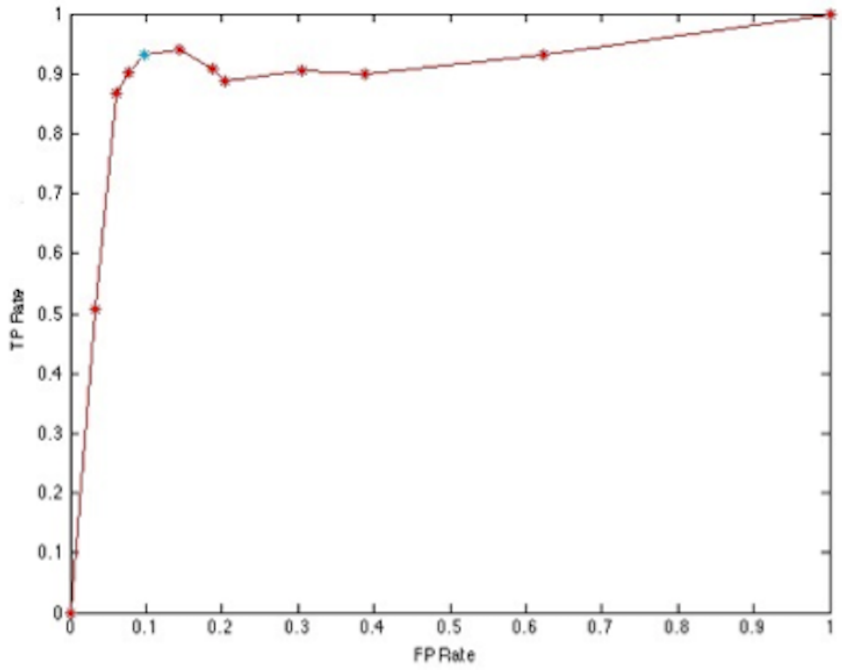

FIGURE 2.1.5: ROC curve for SVM-RBF (Gamma = 0.01, 0.1, 1, 10, 100, 1000, 5000, 10000, 20000, 100000 and Cost = 10). The blue star represents the best result, SVM-RBF (Gamma = 5000 and Cost = 10), with Area under ROC = 0.9165.

2.2 Prediction of Protein Interactions Using SLiMs

In this section, we review a papers related to prediction of protein interactions using SLiMs. The paper has proposed a model that uses SLiMs as properties to predict obligate and non-obligate protein interaction complexes.

2.2.1 Predict obligate and non-obligate protein interaction complexes using SLiMs

The authors of [29] have modeled the prediction problem using SLiMs to extract information contained in the protein sequences to distinguish between obligate and non-obligate PPIs. The authors focus on the problem of determing the transitions from non-obligate (less stable, or transient) to obligate (more stable) complexes. Obligate interactions are permanent, while non-obligate interactions can be permanent or transient [24]. The model delivers classification accuracies as high as 99% on two well-known datasets. Analysis and

cross-dataset validation show that the information contained in the training sequences is crucial for prediction and determination of stability in PPIs.

Previous work and shortcomings by others referred to by the authors

The authors state that characterizing the properties of protein interaction types can be done by studying their sequence or structural information. The most effective approaches for prediction of PPIs use mainly structural information of protein complexes to calculate the feature values, and the PDB is the main source of the molecular structures of protein complexes.

However, models based on structural information from the PDB are not perfect yet and are time comsuming. In addition, the small number of proteins and their interactions from a small number of structures in PDB are very small compared to the number of possible protein interactions available in high-throughput protein and PPIs databases such as UniProt. Moreover, models based on protein structures are limited to availability of structural information. Overall, the authors gained the motivation of a model that can replace the use of structural properties.

The new idea which the authors invented

In that paper, the authors proposed a model that uses SLiMs as properties to predict obligate and non-obligate protein complexes. The model uses k-NN, linear dimensionality reduction (LDR) and SVM as the classifiers to predict these types. Their prediction results for two well-known datasets show prediction accuracy of more than 99%, which implies an increase of at least 7% from previous approaches, even beter than the state-of-art structure-based methods and using only sequence information.

Materials and methods

The authors use two pre-classified datasets of obligate and non-obligate protein complexes from the studies of [46] and [23] as ZH and MW datasets respectively. The ZH dataset contains 75 obligate and 62 non-obligate complexes, and the MW dataset contains 115

obligate and 212 non-obligate complexes.

The authors chose MEME to find independent sets of SLiMs for the two datasets. They optimized the parameters of MEME to find 1,000 SLiMs in both the ZH and MW datasets. They set the length of the SLiMs to 3-10 and 2-7, the minimum number of sites to 8 and the maximun number of sites to 200. Based on the two length ranges of the SLiMs and the two datasets, four SLiM sets were compiled.

The authors indicate that for each complex in the dataset, its sequences are divided into overlapping $l\text{-}mer$, which are the sites of motifs in the training set. Given a sequence X of length L, let us consider an $l\text{-}mer$ a in the sequence, where l is the length of each SLiM. The scoring function they used for processing the scores of the motifs is shown in Formula 2.2.1:

$$I(a|\ X) = -\sum_{i=1}^{l} P(a_i) \times \log(P(a_i)) \tag{2.2.1}$$

where X is the profile sequence, $P(a_i)$ is the probability (of the i^{th} residue of a) from that profile. Since $P(a_i) \leq 1$, $\sum_{i=1}^{l} P(a_i)$ is very small for large sites, -log gives a more meaningful measure.

Equation (2.2.1) implies that that larger the site is, the larger the information content is. Thus, in that paper, they also divide the total information content by the length of the site, l in order to erase the effect. Then the information content of a site a of length l is defined as:

$$\hat{I}(a|\ X) = -\frac{1}{l} \times \sum_{i=1}^{l} P(a_i) \times \log(P(a_i)) \tag{2.2.2}$$

Since log(1) = 0, for any $P(a_i)$ =1, a small threshold is subtracted from $P(a_i)$ as follows:

$$\log P(a_i) = \begin{cases} log(0.99) & \text{if } P(a_i) = 1 \\ log(P(a_i)) & \text{otherwise} \end{cases} \tag{2.2.3}$$

Experiments and analysis

They used two validation approaches for classification. (1) Leave-one-out validation with a k-NN classifier, (2) a cross-dataset validation for testing the accuracy and significance of the newly proposed features. They also chose SVM and LDR for cross-dataset validation classification because the power of generalization of the scheme in prediction of new complexes is provided by SVM and LDR. They used LibSVM with a linear kernel with default parameters for SVM.

Results that the authors claim to have achieved

As for the results of leave-one-out validation with k-NN, the highest accuracy is 98.54% for k = 35 and the lowest is 95.62% for k = 5 when l = 10. This scheme yields the highest accuracy of 99.27% when l = 9, 7, 6, 5. For the ZH dataset, the highest accuracy is 99.27% for different values of l and k. For l = 7, and all the values of k, the accuracy is 99.27%. As for the results of cross-dataset validation, the scheme yields the highest accuracy of 97.81% and 99.27% for l = 5,4 respectively using SVM and different LDR for the ZH dataset with the MW SLiMs for training. As in [46], they predicted obligate and non-obligate complexes with 88.32% accuracy.

The authors note the importance of using SLiMs in prediction of obligate and non-obligate complexes. According to their experiments and results, we considered using SLiMs in prediction of PPIs and CaM-binding proteins. The scoring method here to determine each short subsequence as potential site of the motif, rather than using the sites output by MEME. We use this scoring method in one of our scoring method variance, which is discussed in next chapter, and we call this method "Sliding Window Scoring". We also chose SVM for classification, since SVM yields the highest accuracies in our study.

2.3 Inspiration from the Previous Works

From the first paper of prediction of PPIs using simple codon pairs, we were inspired by the idea of prediction of PPIs using information in the protein sequences, and thus we experimented predicting PPIs using the difference of single amino acids between pairs of proteins. As a result, we not just focus on single amino acids, as we considered 1-gram. We have the idea of enlarging the gram to 2-grams, 3-grams, or even n-grams, and hence, after the consideration, we used SLiMs. Especially in the paper [29], it showed a strong power of SLiMs in prediction of obligate and non-obligate proteins.

Since in the second paper, the simple encoding of protein pairs as difference vectors of amino-acid frequencies between protein pairs yield excellent accuracy results when using kNN, RF or SVM-RBF classifiers, we also chose these classifiers in the experiments presented in this thesis. We chose the same positive and negative protien datasets as mentioned in the paper for prediction of PPIs, although we only randomly chose small parts of them because MEME runs very slow when the input datasets are large.

We use the scoring functions mentioned in the paper [29] for scoring the sites in three scoring method in this thesis because the functions make the position-specific possibility value of each amino acid in the SLiMs has more meaning, also it helps to obtain high accuracies. We also tried the scoring method in our fifth method.

In the three papers, they all chose cross-validation when doing classification. Therefore, in this thesis, we also use cross-validation. Since we deal with much smaller datasets compared to the datasets in the papers, we use 3-fold cross-validation.

CHAPTER 3

Materials and Methods

In this chapter, we describe the datasets and method for the experiments in this thesis. Figure 3.0.1 shows a schematic diagram that depicts our method. First of all, we obtain the positive and negative datasets from the protein databases, and download the protein sequences on Uniprot. Then, we obtain the SLiMs in two different ways, CM and SM, and thereafter we use scoring method with different scoring functions for scoring the sites. Finally we apply Feature Selection and classification to the score matrices and analyze the results.

3.1 Datasets

3.1.1 Datasets for prediction of PPIs

For training the classifiers using machine learning methods we collected positive interaction pairs as well as negative ones. The positive reference set used in our dataset was obtained from the *PrePPI* (structure-based prediction of protein-protein interactions) database, from which we downloaded the *New Human Protein Interaction Set* to create the positive class. The negative reference set was obtained from the *Negatome Database version 2.0* [7], which is a repository of non-interacting pairs of proteins. The corresponding protein sequences were downloaded from Uniprot.

Since MEME runs slow on large datasets, we randomly chose 50 protein pairs from the positive reference set as the positive dataset, and 38 protein pairs from the negative reference set as the negative dataset. All these sequences contain low similarity with other sequences in each dataset.

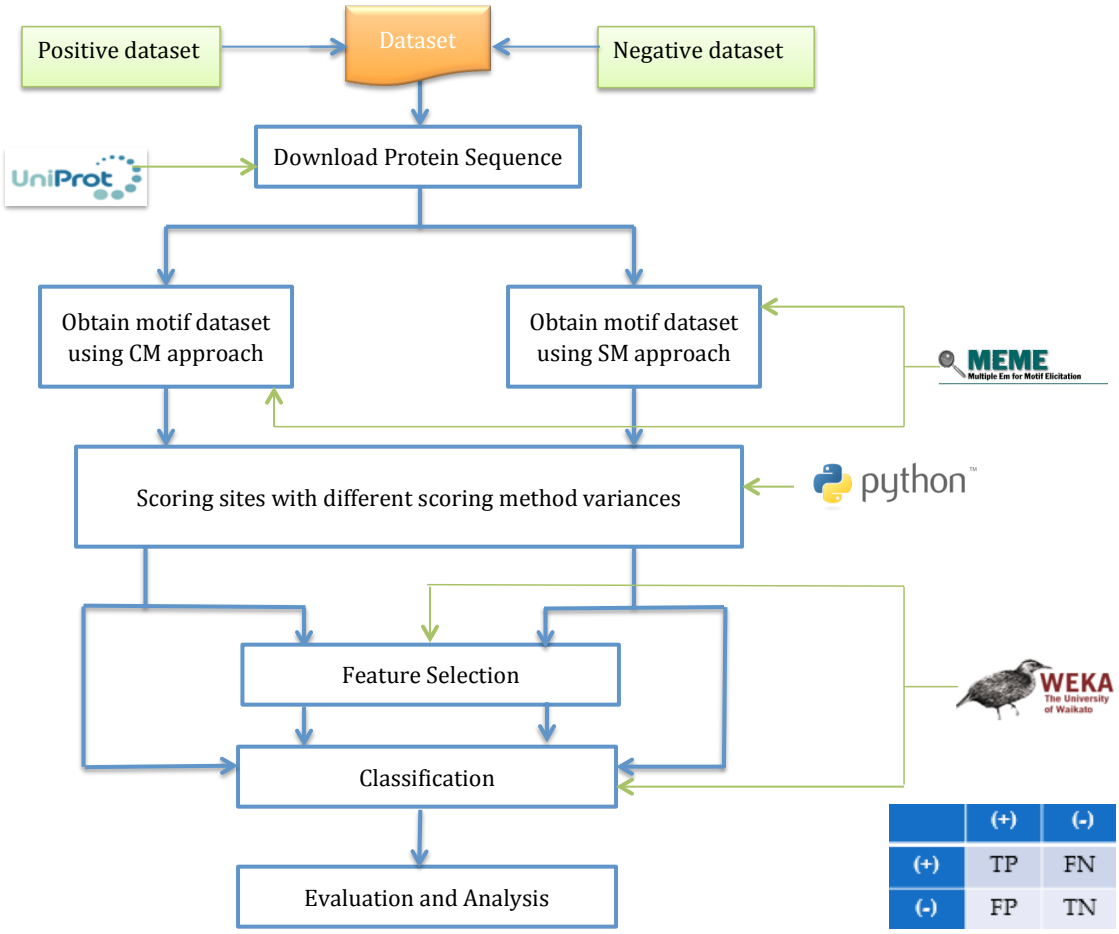

FIGURE 3.0.1: Diagram of the proposed model.

3.1.2 Datasets for prediction of CaM-binding proteins

Our manually curated dataset, which contains 194 CaM-binding proteins collected from the Calmodulin Target Database [43], is used as the positive dataset and 193 Mitochondrial proteins obtained from the Uniprot database as the negative dataset. No major biochemical function has been demonstrated for CaM in the mitochondria. This suggests that the number of CaM-interacting proteins that are localized in the mitochondria will be small relative to other sub-cellular locations. Therefore, we chose proteins that are localized to the mitochondria as our negative dataset. To obtain a more refined list of mitochondrial proteins to use as a negative dataset, all 7,433 proteins that were under the cellular component term Mitochondrion (GO:0005739) and had human taxonomy were downloaded. After filtering out non-reviewed proteins and any proteins with Golgi and Nucleus, 886 proteins were obtained that are strictly mitochondrial as far as GO annotations are concerned. From those remaining Mitochondrial proteins, 193 proteins were selected randomly as the negative dataset, yielding a balanced dataset.

3.2 SLiMs Finding Approaches

We have used MEME to find SLiMs for the datasets. Two different approaches have been used to discover new motifs using MEME: (a) find SLiMs from each of the positive and negative datasets separately (SM) and (b) find SLiMs from the combined positive and negative datasets (CM).

For obtaining the SLiMs datasets for prediction of PPIs, we applied SM using MEME to find 50 SLiMs for each of the positive and negative datasets, separately, obtaining a set of 100 motifs. Similarly, we applied CM to generate 50 SLiMs from the combined positive and negative datasets. The length of the SLiMs were set to a minimum of 3 and a maximum of 10.

As for the SLiMs datasets for prediction of CaM-binding proteins, in the SM approach, we have used MEME to find 50 SLiMs for each of the positive and negative datasets and built a totally 100 SLiMs dataset for the experiments. In the CM approach, we obtained 100 SLiMs from the combined positive and negative datasets. The length of the SLiMs

were also set to a range from 3 to 10.

3.3 Scoring the Sites

Once the SLiM sets are obtained, MEME outputs files that contain patterns of the SLiMs, sites found in the protein sequences and their positions, and the probability matrix of the features of each SLiM. Figure 3.3.1 shows SLiM No.29 found in the CM dataset as output by MEME with the sites found in the sequences and the corresponding protein names. Table 1 shows the *Position-specific probability matrix* (*PSPM*) of this SLiM. The columns represent the 20 amino acids, while the rows correspond to the scores of the features in this SLiM.

We propose five different scoring method variances in this section. Counting sites with different scoring functions and a new approach for defining sites, which we call Sliding Window Scoring method.

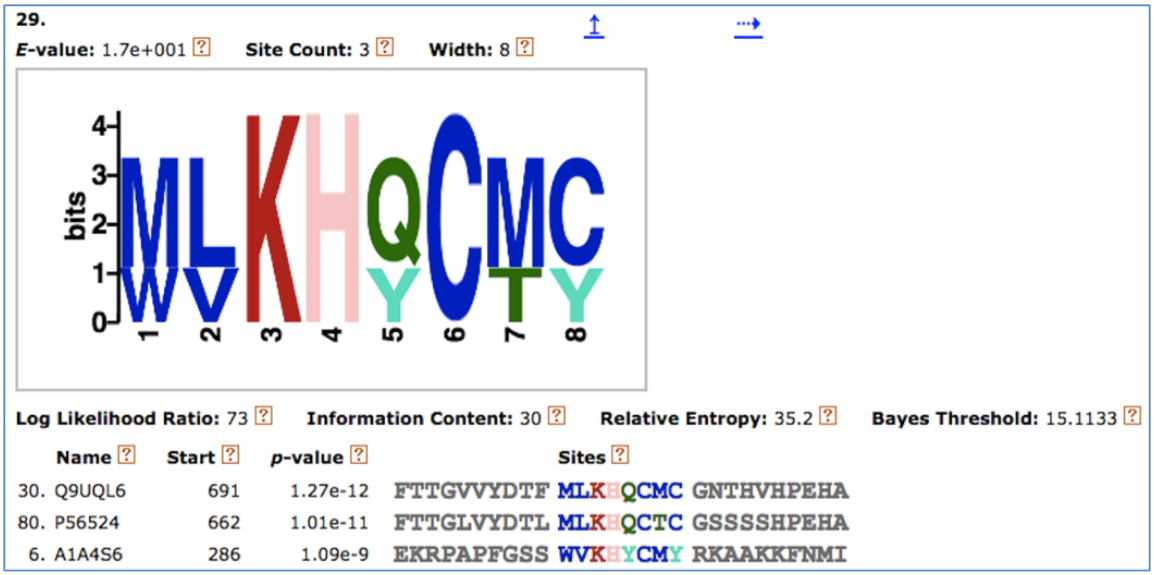

FIGURE 3.3.1: SLiM No.29 found in the CM dataset.

3.3.1 Scoring method variance 1: Counting sites

The first method variance we applied for obtaining the score matrices of proteins is simply counting the numbers of the sites of the correspongding SLiMs appear in the protein

TABLE 3.3.1: Position-specific probability matrix of SLiM No.29.

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	0.6	0	0	0	0	0	0	0	0.3	0
0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0.3	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	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0.3
0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0.3	0	0	0
0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3

sequences. As shown in Figure 3.3.2 , Motif_5 appears three times on the sequence of protein $A0JLT2$ as three sites being output by MEME, and hence the score of Motif_5 for $A0JLT2$ is set to 3. Similarly, Motif_16 appears once in the sequence of the same protein as a site, so the score of Motif_16 for A0JLT2 is set to 1. We applied this scoring method for every protein from the prediction of PPIs.

3.3.2 Scoring method variance 2: Counting sites with I formula

After obtaining good experimented results from the scoring method variance 1 mentioned above, we consider using scoring functions instead of simply counting the SLiMs. *Rueda et al.* [29] used SLiMs as properties for prediction of obligate and non-obligate protein interaction complexes. Their prediction results for two datasets showed an impressive accuracy of more than 99% based on classifiers such as *k-NN*, *LDR* and *SVM*. In that paper, the authors indicate that given a sequence X of length L, they consider an l-mer a in the sequence as a potential site, where l is the length of each SLiM. The scoring function they used for processing the scores of the motifs is as follows, which we call I formula. Thus,

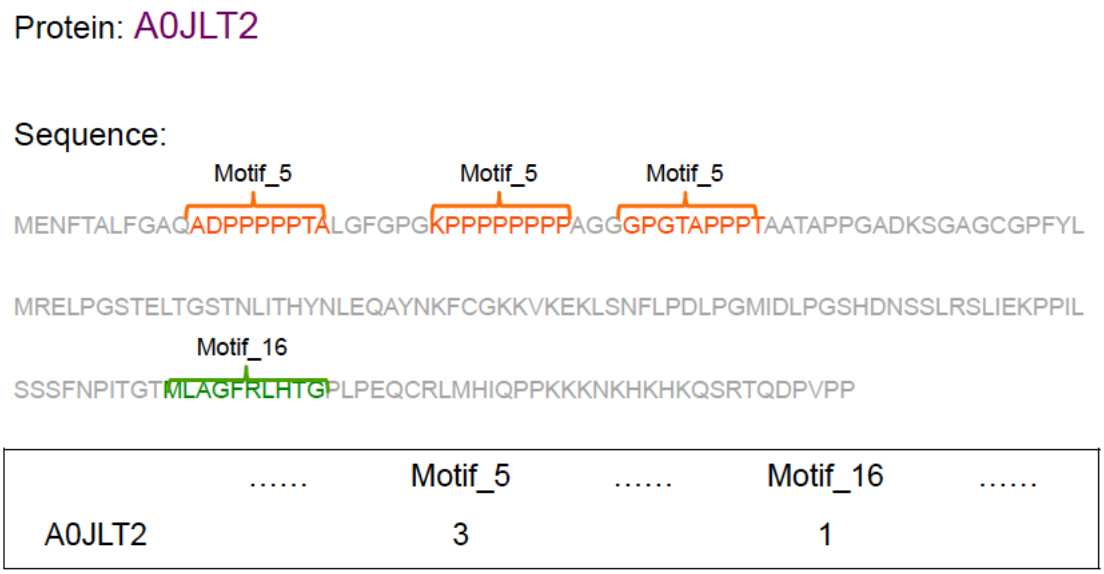

FIGURE 3.3.2: Example of obtaining scores using method variance 1 (Counting SLiMs).

we consider to use this I formula as the scoring function for processing the scores of our SLiMs.

$$I(a|\ X) = -\sum_{i=1}^{l} P(a_i) \times \log(P(a_i)) \tag{3.3.1}$$

where X is the profile sequence, $P(a_i)$ is the probability (of the i^{th} residue of a) from that profile. Since $P(a_i) \leq 1$, $\sum_{i=1}^{l} P(a_i)$ is very small for large sites, so they take -log for a more meaningful measure [29].

Since log(1) = 0, for any $P(a_i)$ =1, a small threshold is subtracted from $P(a_i)$ as follows [29]:

$$\log P(a_i) = \begin{cases} log(0.99) & \text{if } P(a_i) = 1 \\ log(P(a_i)) & \text{otherwise} \end{cases} \tag{3.3.2}$$

Figure 3.3.3 shows an example of obtaining scores using this method variance in our method. Given a sequence profile of protein Q9UQL6, one site of the SLiM No.29 that has been found by MEME is MLKHQCMC. Based on the position-specific probability matrix,

we can score this site using the I formula. In the position-specific probability matrix, each line indicates the scores of corresponding features on each position. Therefore, the score of site MLKHQCMC can be calculaed as $I = -(0.6 \times log(0.6) + 0.6 \times log(0.6) + 1 \times log(0.99) + 0.6 \times log(0.6) + 1 \times log(0.99) + 0.6 \times log(0.6) + 0.6 \times log(0.6))$. We applied this scoring method variance for every protein from the prediction of PPIs.

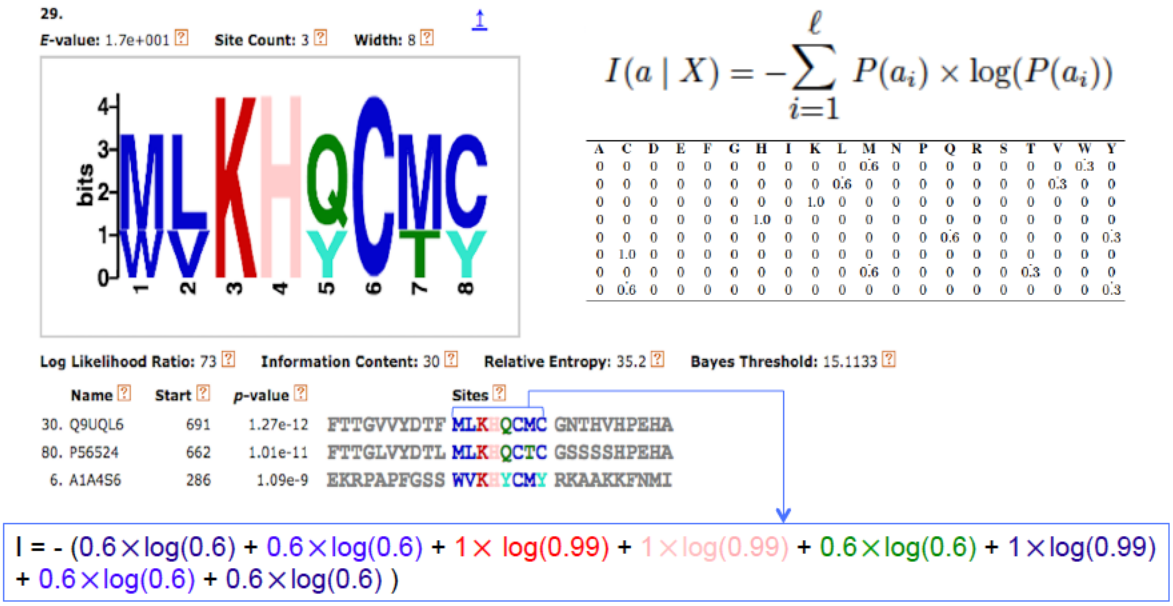

FIGURE 3.3.3: Example of obtaining scores using method variance 2 (Counting SLiMs with I formula).

3.3.3 Scoring method variance 3: Counting sites with $\hat{I}$ formula

Equation (3.3.1) implies that that larger the site is, the larger the information content is. Thus, in [29], the authors also divide the total information content by the length of the site, l in order to erase that effect. Then the information content of a site a of length l is defined as:

$$\hat{I}(a|\ X) = -\frac{1}{l} \times \sum_{i=1}^{l} P(a_i) \times \log(P(a_i)) \tag{3.3.3}$$

We also set the threshold using Equation (3.3.2) to avoid the log(1) = 0 effect. We

applied this scoring method for every protein from both of the prediction of PPIs and prediction of CaM-binding proteins.

3.3.4 Scoring method variance 4: Counting sites with $\hat{I}$ formula / counting of sites

We consider the length of the sites in the scoring method variance 3, and after we obtained all the scores using this scoring method, we infer that the counting of the sites may affect. In order to see the influence of the counting of the sites using the $\hat{I}$ formula, we divided the $\hat{I}$ formula by the counting of corresponding SLiM. Since we already obtained the score matrix of the counting of sites, here we divide the element in the matrix, which obtained by method variance 3, by the element in the other matrix, which obtained by method variance 1, in the same position in their own matrix. We applied this scoring method for every protein from both of the prediction of PPIs and prediction of CaM-binding proteins.

3.3.5 Scoring method 5: Sliding Window Scoring method

After we obtained all of the score matrices using different scoring method variances mentioned above, we thought about a new way to define a site. Thus, we did not consider the sites in the sequences found by MEME. In contrast, we consider every possible subsequence (l-mer) in a sequence as a potential site for a motif of the training set. Each sequence is divided into overlapping l-$mers$. We designed a *Sliding Window Scoring* (*SWS*) method for scoring these sites. Figure 3.3.4 shows an example of SWS based on SLiM No.29 along with its position-specific probability matrix. Let us consider l-mer a in a sequence of length L. We divide the sequence into all possible overlapping l-$mers$ of length W, where l is the length of each SLiM, and deliver a total of $\{L - W + 1\}$ l-$mers$. Then, Equation (3.3.4) is used to calculate the information contained in l-mer a, given a profile X of length L, and a SLiM m of length W:

$$P(a|\ X) = \sum_{i=1}^{W} P(a_i) \tag{3.3.4}$$

where X is the profile of the sequence, $P(a_i)$ is the probability of the amino acid in that profile. Since $P(a|\ X)$ may be 0 or very small if the SLiM and the site have very low similarity, we set a threshold to 60% for $P(a|\ X)$. Thus, we do not consider this site and remove the $P(a|\ X)$ score as well. Once the scores for all possible $l\text{-}mers$ in profile X are obtained, we use Equation (3.3.5) to add up all the scores of the $l\text{-}mers$ as the score of SLiM m for profile X:

$$P(m|\ X) = \sum_{i=1}^{L-W+1} P(a|\ X) \tag{3.3.5}$$

Equation (3.3.5) implies that the more likely a is a site, the larger the information content is. Thus, in order to erase this effect, we also divide the total information content by the number of sites in the sequence, $N \leqslant L - W + 1$, since we removed some site with scores lower than 60%):

$$\hat{P}(m|\ X) = \frac{1}{N} \times \sum_{i=1}^{L-W+1} P(a|\ X) \tag{3.3.6}$$

Then, we calculate $P(m|\ X)$ and $\hat{P}(m|\ X)$ for all the SLiMs obtained from both CM and SM datasets for each protein sequence. We applied this scoring method for every protein from both the prediction of PPIs and prediction of CaM-binding proteins.

3.4 Score Processing

After we obtained all the score matrices, we need to process the scores seperately since the proteins for prediction of PPIs are in pairs, while the proteins for prediction of CaM-binding

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	0.6	0	0	0	0	0	0	0	0.3	0
0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0.3	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	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0.3
0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0.3	0	0	0
0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3

Protein sequence: MLKHQCMCGNTHVHPEH

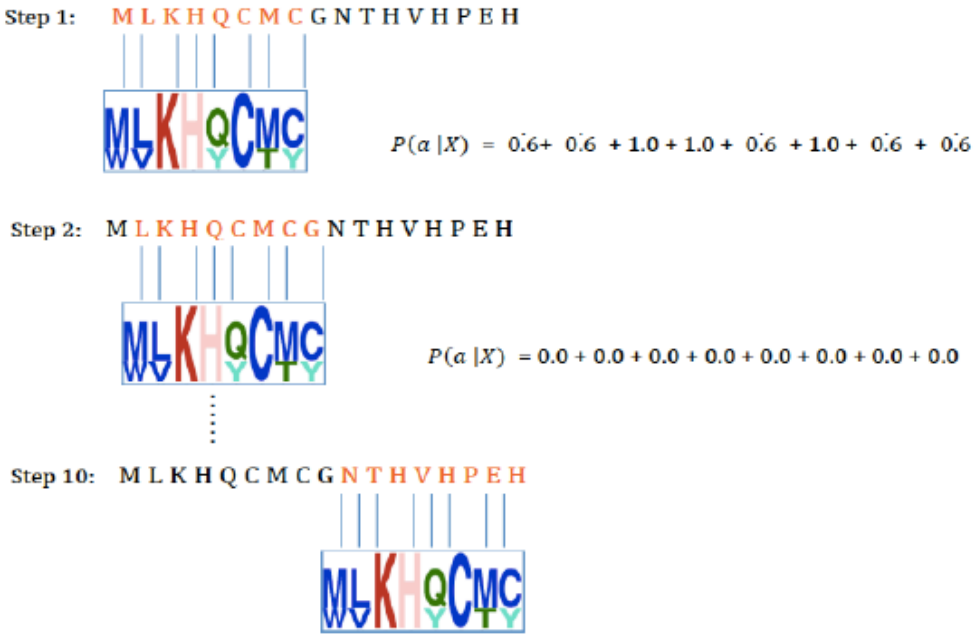

FIGURE 3.3.4: Example of the SWS method based on SLiM No.29 along with its position-specific probability matrix.

are single proteins. Figure 3.4.1 shows an example of score processing for prediction of PPIs and CaM-binding proteins.

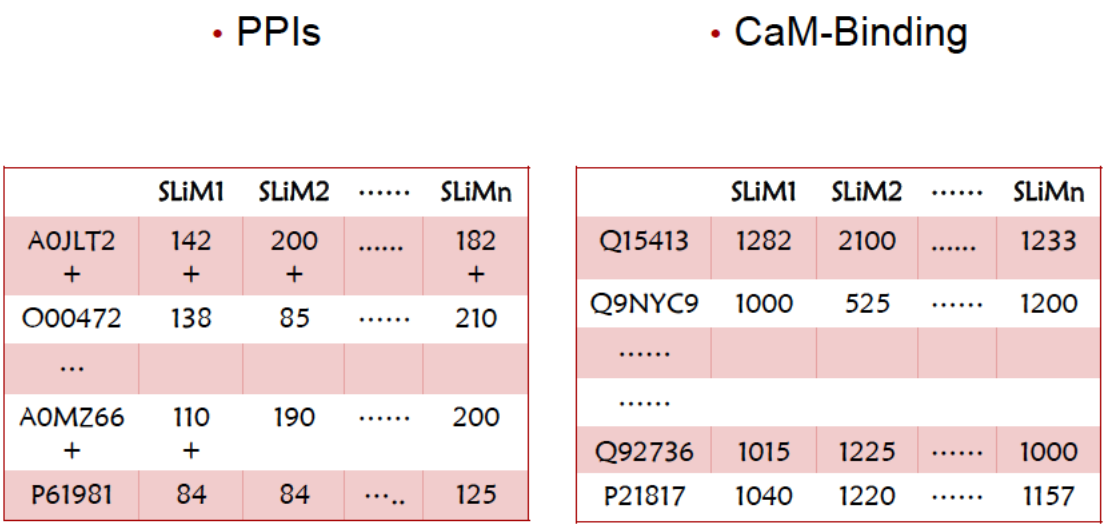

FIGURE 3.4.1: Example of score processing for prediction of PPIs and CaM-binding proteins.

3.4.1 Score processing for prediction of PPIs

For prediction of PPIs using the SWS method, considering we are given a protein pair (P_i, P_j), s_{i1} to s_{in} are the $P(m|\ X)$ scores of each SLiM in sequence P_i, while s_{j1} to s_{jn} are the $P(m|\ X)$ scores of each SLiM on sequence of protein P_j, n is the number of SLiMs, so P_i = s_{i1}, s_{i2}, ... , s_{in} and P_j = s_{j1}, s_{j2}, ... , s_{jn}. Also, we use the $\hat{P}(m|\ X)$ values of each SLiM on sequences P_i and P_j respectively to generate t_{i1} to t_{in} and t_{j1} to t_{jn}, where P_i = t_{i1}, t_{i2}, ... , t_{in} and P_j = t_{j1}, t_{j2}, ... , t_{jn}. Thus, pair (P_i, P_j) is transformed into two total score vectors of length n as follows:

$$S_{ij} = (s_{i1} + s_{j1}, ..., s_{in} + s_{jn})$$

$$T_{ij} = (t_{i1} + t_{j1}, ..., t_{in} + t_{jn})$$

where S_{ab} and T_{ab} are the $P(m|\ X)$ and $\hat{P}(m|\ X)$ scores of SLiM b of the sequence of protein P_a, S_{ij} is the total score matrix of $P(m|\ X)$ scores for n SLiMs in each protein sequence pair, and T_{ij} is the total score matrix of $\hat{P}(m|\ X)$ scores for the same n SLiMs in each protein sequence pair. We call the matrices S score matrix and T score matrix for S_{ij} and T_{ij} respectively. This transformation is applied to each positive pair and each negative pair in the training set with the SLiMs obtained from both CM and SM approaches.

Similarly, for prediction of PPIs using method variance 1 to method variance 4, , we also add up the scores between pairs of proteins as the score for each SLiM, and applied this transformation to each pair with the SLiMs obtained from both CM and SM.

3.4.2 Score processing for CaM-binding proteins

As for prediction of CaM-binding proteins using the SWS method, since they are single proteins, we determine that s_{i1} to s_{in} are the $P(m|\ X)$ scores of each SLiM on every sequence of the protein, while t_{i1} to t_{in} are the $\hat{P}(m|\ X)$ scores of each SLiM on every sequence of the protein, where n is the frequency of SLiMs. Thus, each protein sequence is transformed into two total score vectors of length n as follows:

$$S_{ij} = (s_{i1}, s_{i2}, ..., s_{in})$$

$$T_{ij} = (t_{i1}, t_{i2}, ..., t_{in})$$

where S_i and T_i are the $P(m|\ X)$ and $\hat{P}(m|\ X)$ scores of SLiM for n SLiMs on each protein sequence. We call the matrices S score matrix and T score matrix for S_i and T_i respectively. This transformation is also applied to each positive pair and each negative pair in the training set with the SLiMs obtained from both the SM and CM approaches.

Similarly, we applied the same score processing for prediction of CaM-binding using method variances 3 and 4.

3.5 Machine Learning Method Using for Classification

We applied *SVM-Polynomial kernel*, *RF*, *kNN*, *DT* and *MP* classifiers on our dataset using *WEKA* ver. 3.7.11 software [1]. We applied all these classifiers with default parameters: k

= 1 for k-NN and $Gamma(g) = 0$ and $Cost(c) = 1$ for SVM + Polynomial kernel.

We also applied FS, and used RF for evaluating the accuracy of a feature subset. 3-fold Cross-Validation is the method we used for training and evaluating all the classifiers. We used Accuracy, Recall and MCC to assess the predictive capability of our approach as mentioned in Chapter 1.

CHAPTER 4

Results

We have applied five different scoring methods for prediction of PPIs and CaM-binding proteins, but the results among all of them are quite similar, thus, in this chapter, we analyze and discuss only one scoring method variance, the SWS method, which obtains the best results. We select the best results among all of the results with 5 different classifiers: SVM-Polynomial with C = 1 and gamma = 0 ($c = 1$, $g = 0$), RF, 1-NN, DT, MP.

4.1 Results

As for the SWS method, we have performed eight sets of experiments for both the prediction of PPIs and CaM-binding proteins: classifying the (1) S and (2) T score matrices with SLiMs obtained from SM, classifying the (3) S and (4) T score matrices using the feature subset selected by FS with SLiMs obtained from SM, classifying the (5) S and (6) T score matrices with SLiMs obtained from CM and classifying the (7) S and (8) T score matrices using the feature subset selected by FS with SLiMs obtained from CM.

4.1.1 Classification results of prediction of PPIs

Table 4.1.1 shows the classification results for the score matrices with SLiMs obtained from the CM dataset while Table 4.1.2 shows the classification results for the score matrices with SLiMs obtained from the SM dataset.

By observing Tables 4.1.1 and 4.1.2, it is noticed that for the SLiMs obtained from the CM dataset, the classification accuracies range from 67.0% to 84.1% among all of the classification experiments, SVM-Polynomial on the T score matrix subset after FS yields

TABLE 4.1.1: Prediction of PPIs classification results for the score matrices with SLiMs obtained from the CM approach.

Dataset for Classification	Classifier	Accuracy (%)	Recall (%)	MCC
S score matrix	SVM-Polynomial $(c = 1, g = 0)$	76.1	76.1	0.525
	Random Forest	76.1	76.1	0.509
	k-NN $(k = 1)$	75.0	75.0	0.488
	Decision Tree	64.8	64.8	0.270
	Multilayer Perceptron	**79.5**	79.5	0.586
T score matrix	SVM-Polynomial $(c = 1, g = 0)$	79.5	79.5	0.583
	Random Forest	73.9	73.9	0.462
	k-NN $(k = 1)$	**81.8**	81.8	0.651
	Decision Tree	67.0	67.0	0.331
	Multilayer Perceptron	78.4	78.4	0.557
S score matrix subset selected by FS	SVM-Polynomial $(c = 1, g = 0)$	78.4	78.4	0.562
	Random Forest	79.5	79.5	0.580
	k-NN $(k = 1)$	**80.7**	80.7	0.611
	Decision Tree	75.0	75.0	0.499
	Multilayer Perceptron	79.5	79.5	0.591
T score matrix subset selected by FS	SVM-Polynomial $(c = 1, g = 0)$	**84.1**	84.1	0.675
	Random Forest	81.8	81.8	0.629
	k-NN $(k = 1)$	81.8	81.8	0.636
	Decision Tree	79.5	79.5	0.581
	Multilayer Perceptron	79.5	79.5	0.583

TABLE 4.1.2: Accuracies of prediction of PPIs classification for the score matrices with SLiMs obtained from the SM approach.

Dataset for Classification	Classifier	Accuracy (%)	Recall (%)	MCC
S score matrix	SVM-Polynomial $(c = 1, g = 0)$	73.9	73.9	0.492
	Random Forest	72.7	72.7	0.439
	k-NN $(k = 1)$	72.7	72.7	0.439
	Decision Tree	58.0	58.0	0.146
	Multilayer Perceptron	**81.8**	81.8	0.632
T score matrix	SVM-Polynomial $(c = 1, g = 0)$	56.8	56.8	0.000
	Random Forest	**78.4**	78.4	0.564
	k-NN $(k = 1)$	77.3	77.3	0.533
	Decision Tree	63.6	63.6	0.244
	Multilayer Perceptron	70.5	70.5	0.390
S score matrix subset selected by FS	SVM-Polynomial $(c = 1, g = 0)$	70.5	70.5	0.392
	Random Forest	**78.4**	78.4	0.558
	k-NN $(k = 1)$	72.7	72.7	0.453
	Decision Tree	77.3	77.3	0.537
	Multilayer Perceptron	77.3	77.3	0.545
T score matrix subset selected by FS	SVM-Polynomial $(c = 1, g = 0)$	56.8	56.8	0.000
	Random Forest	75.0	75.0	0.486
	k-NN $(k = 1)$	79.5	79.5	0.587
	Decision Tree	75.0	75.0	0.491
	Multilayer Perceptron	**80.7**	80.7	0.604

the highest classification accuracies, ranging from 76.1% to 84.1%. For the SLiMs obtained from the SM dataset, Multilayer Perceptron on the S score matrix yields the highest classification accuracy, 81.8%.

4.1.2 Grid search for SVM-polynomial (prediction of PPIs)

We applied grid search using SVM Polynomial on prediction of PPIs for four kinds of matrices datasets as shown in Tables 4.1.3 and 4.1.4 with SLiMs obtained from SM and CM separately, with different values of parameter C = 1, 10, 100, 1000, gamma = 0.01, 0.1, 0, 1, 10, 100, 1,000. We chose 3-fold cross-validation for evaluation.

Observing to Tables 4.1.3 and 4.1.4, we find that after applying grid search for SVM-Polynomial kernel with different values of C and gamma, the accuracy goes up to 84.1% with SLiMs obtained from the CM dataset and it reaches 86.4% with SLiMs obtained from the SM dataset. This means that the value of the parameters plays an important role in our approach.

4.1.3 Classification results of prediction of CaM-binding proteins

Table 4.1.5 shows the classification results for the score matrices with SLiMs obtained from SM while Table 4.1.6 shows the classification results for the score matrices with SLiMs obtained from CM of CaM-binding proteins using the SWS method.

By observing Tables 4.1.5 and 4.1.6, it is noticed that for the SLiMs obtained from SM, 1-NN on the S score matrix yields the highest classification accuracy of 80.6%. For the SLiMs obtained from CM, the classification accuracies range from 57.6% to 80.1% among all of the classification experiments. RF on the S score matrix subset after FS yields the highest classification accuracies, ranging from 69.3% to 80.1%.

TABLE 4.1.3: Accuracies (%) of prediction of PPIs using SVM-Polynomial (C = 1, 10, 100, 1000, gamma = 0.01, 0.1, 0, 1, 10, 100, 1000) with SLiMs obtained from SM.

		C=1	C=10	C=100	C=1,000
S score matrix	gamma=0	61.4	61.4	61.4	61.4
	gamma=0.01	61.4	61.4	61.4	61.4
	gamma=0.1	61.4	61.4	61.4	61.4
	gamma=1	61.4	61.4	61.4	61.4
	gamma=10	61.4	61.4	61.4	61.4
	gamma=100	61.4	61.4	61.4	61.4
	gamma=1,000	61.4	61.4	61.4	61.4
T score matrix	gamma=0	56.8	56.8	60.2	71.6
	gamma=0.01	56.8	56.8	60.2	71.6
	gamma=0.1	56.8	60.2	71.6	76.1
	gamma=1	59.1	71.6	76.1	71.6
	gamma=10	71.6	76.1	72.7	70.5
	gamma=100	78.4	78.4	79.5	79.5
	gamma=1,000	62.5	63.6	63.6	63.6
S score matrix subset selected by FS	gamma=0	61.4	61.4	61.4	61.4
	gamma=0.01	63.6	63.6	63.6	63.6
	gamma=0.1	61.4	61.4	61.4	61.4
	gamma=1	61.4	61.4	61.4	61.4
	gamma=10	61.4	61.4	61.4	61.4
	gamma=100	61.4	61.4	61.4	61.4
	gamma=1,000	61.4	61.4	61.4	61.4
T score matrix subset selected by FS	gamma=0	56.8	56.8	58.0	69.3
	gamma=0.01	56.8	56.8	56.8	56.8
	gamma=0.1	56.8	56.8	56.8	69.3
	gamma=1	56.8	56.8	69.3	78.4
	gamma=10	56.8	69.3	**84.1**	83.0
	gamma=100	72.7	**84.1**	79.5	78.4
	gamma=1,000	75.0	76.1	68.2	65.9

TABLE 4.1.4: Accuracies (%) of prediction of PPIs using SVM-Polynomial (C = 1, 10, 100, 1000, gamma = 0.01, 0.1, 0, 1, 10, 100, 1000) with SLiMs obtained from CM.

		C=1	C=10	C=100	C=1,000
S score matrix	gamma=0	76.1	76.1	76.1	76.1
	gamma=0.01	76.1	76.1	76.1	76.1
	gamma=0.1	76.1	76.1	76.1	76.1
	gamma=1	76.1	76.1	76.1	76.1
	gamma=10	76.1	76.1	76.1	76.1
	gamma=100	76.1	76.1	76.1	76.1
	gamma=1,000	76.1	76.1	76.1	76.1
T score matrix	gamma=0	79.5	78.4	77.3	77.3
	gamma=0.01	68.2	79.5	76.1	77.3
	gamma=0.1	77.3	77.3	77.3	77.3
	gamma=1	77.3	77.3	77.3	77.3
	gamma=10	77.3	77.3	77.3	77.3
	gamma=100	77.3	77.3	77.3	77.3
	gamma=1,000	77.3	77.3	77.3	77.3
S score matrix subset selected by FS	gamma=0	78.4	78.4	78.4	78.4
	gamma=0.01	78.4	78.4	78.4	78.4
	gamma=0.1	78.4	78.4	78.4	78.4
	gamma=1	78.4	78.4	78.4	78.4
	gamma=10	78.4	78.4	78.4	78.4
	gamma=100	78.4	78.4	78.4	78.4
	gamma=1,000	78.4	78.4	78.4	78.4
T score matrix subset selected by FS	gamma=0	84.1	84.1	**86.4**	84.1
	gamma=0.01	56.8	56.8	60.2	85.2
	gamma=0.1	85.2	84.1	85.2	**86.4**
	gamma=1	**86.4**	80.7	78.4	77.3
	gamma=10	79.5	79.5	79.5	79.5
	gamma=100	75.0	75.0	75.0	75.0
	gamma=1,000	79.5	79.5	79.5	79.5

TABLE 4.1.5: Prediction of CaM-binding proteins classification results for the score matrices with SLiMs obtained from SM.

Dataset for Classification	Classifier	Accuracy (%)	Recall (%)	MCC
S score matrix	SVM-Polynomial $(c = 1, g = 0)$	72.6	72.6	0.453
	Random Forest	73.1	73.1	0.463
	k-NN $(k = 1)$	**80.6**	80.6	0.612
	Decision Tree	72.9	72.9	0.466
	Multilayer Perceptron	76.0	76.0	0.533
T score matrix	SVM-Polynomial $(c = 1, g = 0)$	55.0	55.0	0.105
	Random Forest	68.5	68.5	0.375
	k-NN $(k = 1)$	59.7	59.7	0.275
	Decision Tree	68.2	68.2	0.364
	Multilayer Perceptron	**75.7**	75.7	0.518
S score matrix subset selected by FS	SVM-Polynomial $(c = 1, g = 0)$	56.1	56.1	0.122
	Random Forest	**77.8**	77.8	0.556
	k-NN $(k = 1)$	77.0	77.0	0.542
	Decision Tree	74.2	74.2	0.495
	Multilayer Perceptron	76.2	76.2	0.545
T score matrix subset selected by FS	SVM-Polynomial $(c = 1, g = 0)$	64.9	64.9	0.297
	Random Forest	**69.3**	69.3	0.385
	k-NN $(k = 1)$	66.4	66.4	0.330
	Decision Tree	66.7	66.7	0.334
	Multilayer Perceptron	68.0	68.0	0.360

TABLE 4.1.6: Prediction of CaM-binding proteins classification results for the score matrices with SLiMs obtained from CM.

Dataset for Classification	Classifier	Accuracy (%)	Recall (%)	MCC
S score matrix	SVM-Polynomial $(c = 1, g = 0)$	72.6	72.6	0.453
	Random Forest	74.7	74.7	0.494
	k-NN $(k = 1)$	**78.3**	78.3	0.566
	Decision Tree	71.3	71.3	0.436
	Multilayer Perceptron	76.5	76.5	0.553
T score matrix	SVM-Polynomial $(c = 1, g = 0)$	57.6	57.6	0.213
	Random Forest	69.3	69.3	0.395
	k-NN $(k = 1)$	58.1	58.1	0.261
	Decision Tree	65.1	65.1	0.303
	Multilayer Perceptron	**71.3**	71.3	0.436
S score matrix subset selected by FS	SVM-Polynomial $(c = 1, g = 0)$	62.0	62.0	0.240
	Random Forest	**80.1**	80.1	0.603
	k-NN $(k = 1)$	78.6	78.6	0.571
	Decision Tree	72.1	72.1	0.455
	Multilayer Perceptron	77.0	77.0	0.560
T score matrix subset selected by FS	SVM-Polynomial $(c = 1, g = 0)$	60.2	60.2	0.210
	Random Forest	**70.5**	70.5	0.415
	k-NN $(k = 1)$	68.7	68.7	0.382
	Decision Tree	68.7	68.6	0.379
	Multilayer Perceptron	68.5	68.5	0.370

4.1.4 Grid search for SVM-polynomial (prediction of CaM-binding proteins)

Similarly, we applied grid search with different values of parameter C = 1, 10, 100, 1000, gamma = 0.01, 0.1, 0, 1, 10, 100, 1000 on prediction of CaM-binding proteins for the score matrices using SVM-polynomial as shown in Tables 4.1.7 and 4.1.8 with SLiMs obtained from SM and CM separately. We chose 3-fold cross-validation for evaluation.

TABLE 4.1.7: Accuracies (%) of prediction of CaM-binding proteins using SVM-Polynomial (C = 1, 10, 100, 1000, gamma = 0.01, 0.1, 0, 1, 10, 100, 1000) with SLiMs obtained from SM.

		C=1	C=10	C=100	C=1,000
S score matrix	gamma=0	72.6	72.6	72.6	72.6
	gamma=0.01	72.6	72.6	72.6	72.6
	gamma=0.1	72.6	72.6	72.6	72.6
	gamma=1	72.6	72.6	72.6	72.6
	gamma=10	72.6	72.6	72.6	72.6
	gamma=100	72.6	72.6	72.6	72.6
	gamma=1,000	72.6	72.6	72.6	72.6
T score matrix	gamma=0	55.0	69.8	**75.2**	73.6
	gamma=0.01	55.0	70.3	**75.5**	73.4
	gamma=0.1	73.4	70.5	68.0	68.7
	gamma=1	68.7	68.7	68.7	68.7
	gamma=10	68.7	68.7	68.7	68.7
	gamma=100	68.7	68.7	68.7	68.7
	gamma=1,000	68.7	68.7	68.7	68.7
S score matrix subset selected by FS	gamma=0	45.0	45.0	45.0	45.0
	gamma=0.01	42.4	69.3	61.5	61.5
	gamma=0.1	62.8	62.8	62.8	62.8
	gamma=1	45.0	45.0	45.0	45.0
	gamma=10	49.9	49.9	49.9	49.9
	gamma=100	48.3	48.3	48.3	48.3
	gamma=1,000	56.1	56.1	56.1	56.1
T score matrix subset selected by FS	gamma=0	53.0	62.5	62.5	62.5
	gamma=0.01	53.0	53.0	53.0	53.0
	gamma=0.1	53.0	62.5	62.5	62.5
	gamma=1	63.0	66.9	67.2	64.6
	gamma=10	66.7	66.4	66.9	65.4
	gamma=100	64.3	63.6	58.4	64.3
	gamma=1,000	58.7	66.4	64.1	61.8

TABLE 4.1.8: Accuracies (%) of prediction of CaM-binding proteins using SVM-Polynomial (C = 1, 10, 100, 1000, gamma = 0.01, 0.1, 0, 1, 10, 100, 1000) with SLiMs obtained from CM.

		C=1	C=10	C=100	C=1,000
S score matrix	gamma=0	72.6	72.6	72.6	72.6
	gamma=0.01	72.6	72.6	72.6	72.6
	gamma=0.1	72.6	72.6	72.6	72.6
	gamma=1	72.6	72.6	72.6	72.6
	gamma=10	72.6	72.6	72.6	72.6
	gamma=100	72.6	72.6	72.6	72.6
	gamma=1,000				
T score matrix	gamma=0	57.6	71.3	**72.9**	71.3
	gamma=0.01	57.6	71.8	**72.9**	71.3
	gamma=0.1	71.3	68.5	70.3	70.3
	gamma=1	70.3	70.3	70.3	70.3
	gamma=10	70.3	70.3	70.3	70.3
	gamma=100	70.3	70.3	70.3	70.3
	gamma=1,000	70.3	70.3	70.3	70.3
S score matrix subset selected by FS	gamma=0	62.0	62.0	62.0	62.0
	gamma=0.01	44.7	44.7	44.7	44.7
	gamma=0.1	52.7	52.7	52.7	52.7
	gamma=1	49.1	49.1	49.1	49.1
	gamma=10	66.9	66.9	66.9	66.9
	gamma=100	55.3	55.3	55.3	55.3
	gamma=1,000	33.6	33.6	33.6	33.6
T score matrix subset selected by FS	gamma=0	60.2	63.3	64.6	65.6
	gamma=0.01	58.1	58.1	58.1	60.2
	gamma=0.1	60.2	63.3	64.6	65.6
	gamma=1	65.6	69.5	73.9	73.1
	gamma=10	70.8	66.7	68.5	69.5
	gamma=100	71.8	69.8	67.4	69.8
	gamma=1,000	66.4	65.6	65.9	63.8

Observing Tables 4.1.7 and 4.1.8, we find that after applying grid search for SVM-Polynomial kernel with different values of C and gamma, the accuracy goes up to 72.9% with SLiMs obtained from CM and it reaches 75.5% with SLiMs obtained from SM. Compared with the results by SVM-Polynomial with C = 1, gamma = 0, the grid search does not improve the classification results.

4.2 Comparison

4.2.1 Comparison between results of prediction of PPIs

Following the classification results shown in Tables 4.1.1 and 4.1.2, we plot all the accuracies among the classification results for the four matrices with SLiMs obtained from CM as shown in Figure 4.2.1. From Figure 4.2.1, we observe that the T score matrix subset selected by FS obtained the highest accuracies on SVM-Polynomial with C = 1 and gamma = 0, Random Forest and Decision Tree classifiers. The original T score matrix achieves the highest accuracy on 1-NN. Most of the accuracies obtained from the T score matrix are higher than the accuracies obtained from the S score matrix among different classifiers. The classifiers perform better after FS on both of the S and T score matrices.

We also compare all the accuracies among the classification results for the four matrices with SLiMs obtained from SM as shown in Figure 4.2.2. The S score matrix yielded the highest accuracies on Multilayer Perceptron, and the T score matrix subset selected by FS also obtained accuracy which is above 80%.

4.2.2 Comparison between results of prediction of CaM-binding proteins results

Similarly, following the classification results shown in Tables 4.1.5 and 4.1.6, we plot all the accuracies among the classification results for the four matrices with SLiMs obtained from CM as shown in Figure 4.2.3. From Figure 4.2.3, we observe that the T score matrix subset selected by FS yielded the highest accuracies on Random Forest. Most of the accuracies obtained from S score matrices are higher than the accuracies obtained from T

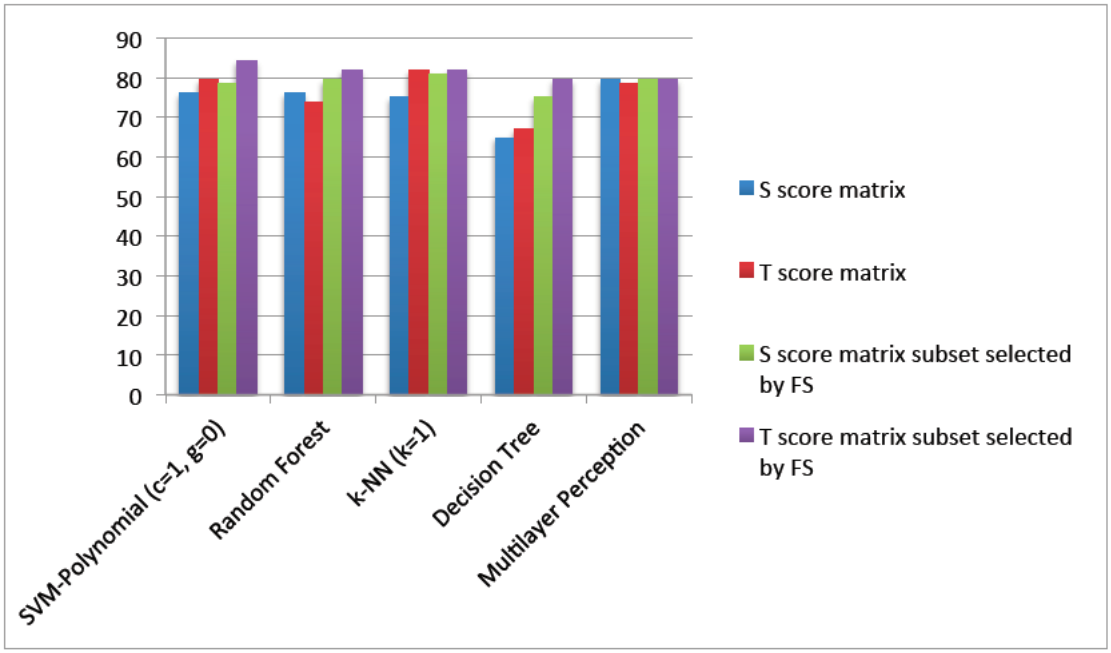

FIGURE 4.2.1: Accuracies for prediction of PPIs for matrices with SLiMs obtained from CM.

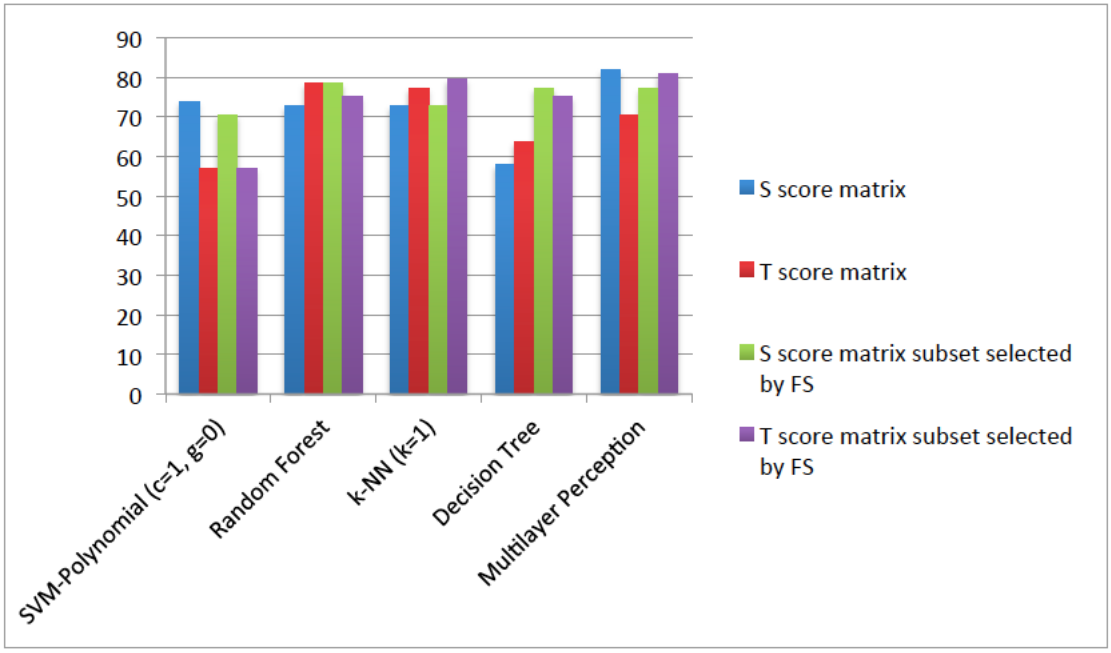

FIGURE 4.2.2: Accuracies for prediction of PPIs for matrices with SLiMs obtained from SM.

score matrices among different classifiers. The classifiers perform better after FS on most of both of the S and T score matrix.

We also compare all the accuracies among the classification results for the four matrices with SLiMs obtained from SM as shown in Figure 4.2.4. The S score matrix obtained the highest accuracies on 1-NN, while all of the accuracies obtained from S score matrices are higher than the accuracies obtained from T score matrices among different classifiers.

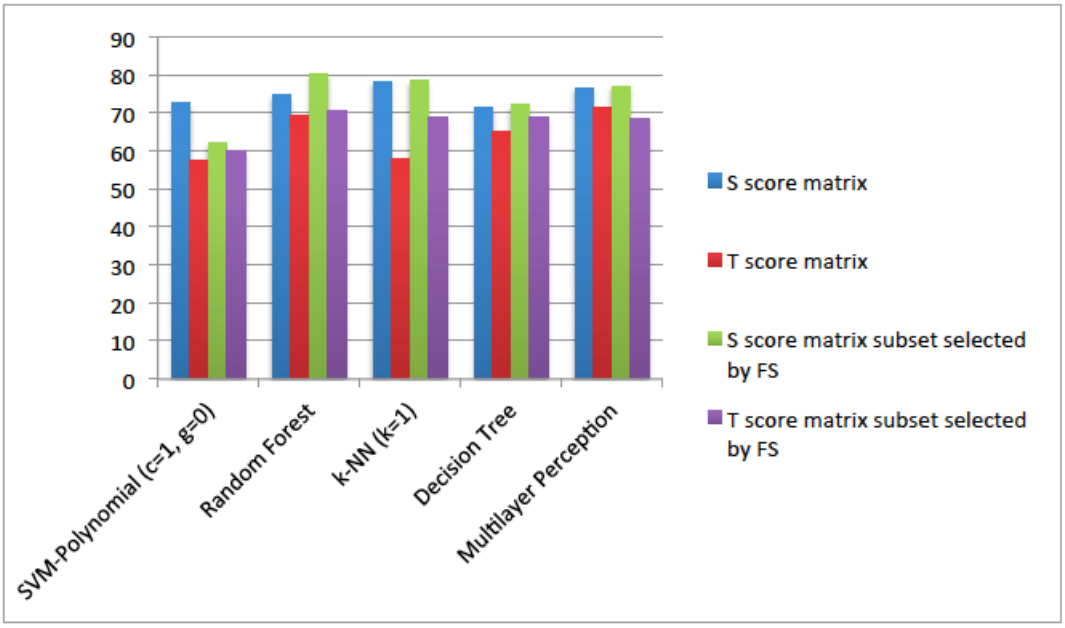

FIGURE 4.2.3: Accuracies for prediction of CaM-binding for matrices with SLiMs obtained from CM.

Table 4.2.5 shows the comparison of prediction of CaM-binding proteins between results of SM and CM, using the 1-NN classifier. Both S and T score matrices yield higher accuracies with SM than the matrices with CM using 1-NN classifier, while after FS, both S and T score matrices yield higher accuracies with CM.

4.2.3 Classification VS Classification + FS

We compared classification results between non-FS and FS, using the classifier that performs generally the best in either prediction of PPIs or prediction of CaM-binding proteins. Thus, here we chose 1-NN and RF classifiers for the comparison.

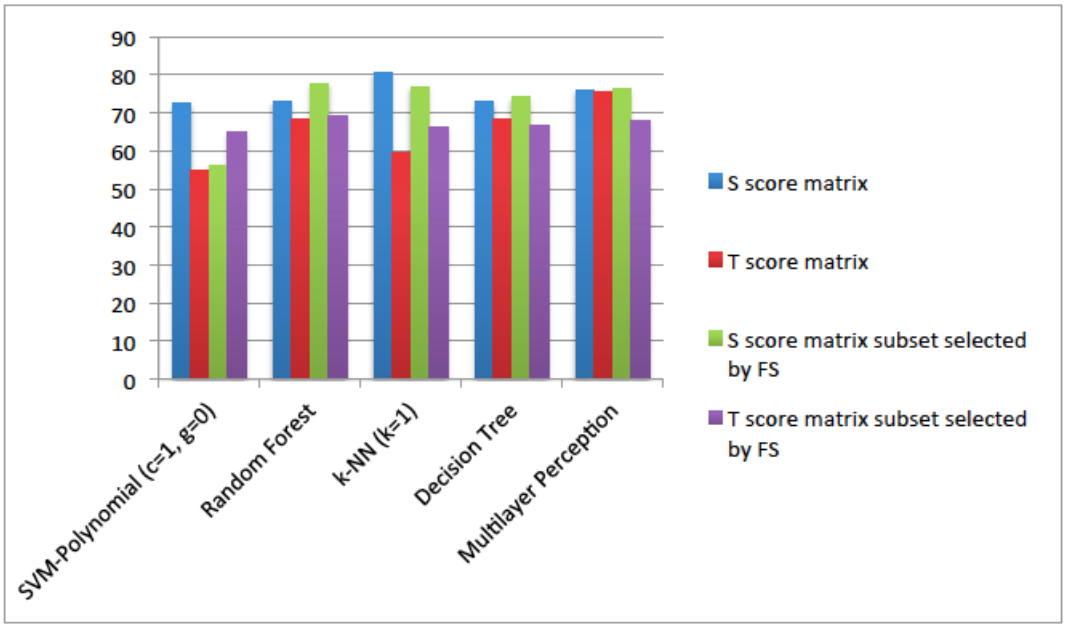

FIGURE 4.2.4: Accuracies for prediction of CaM-binding for matrices with SLiMs obtained from SM.

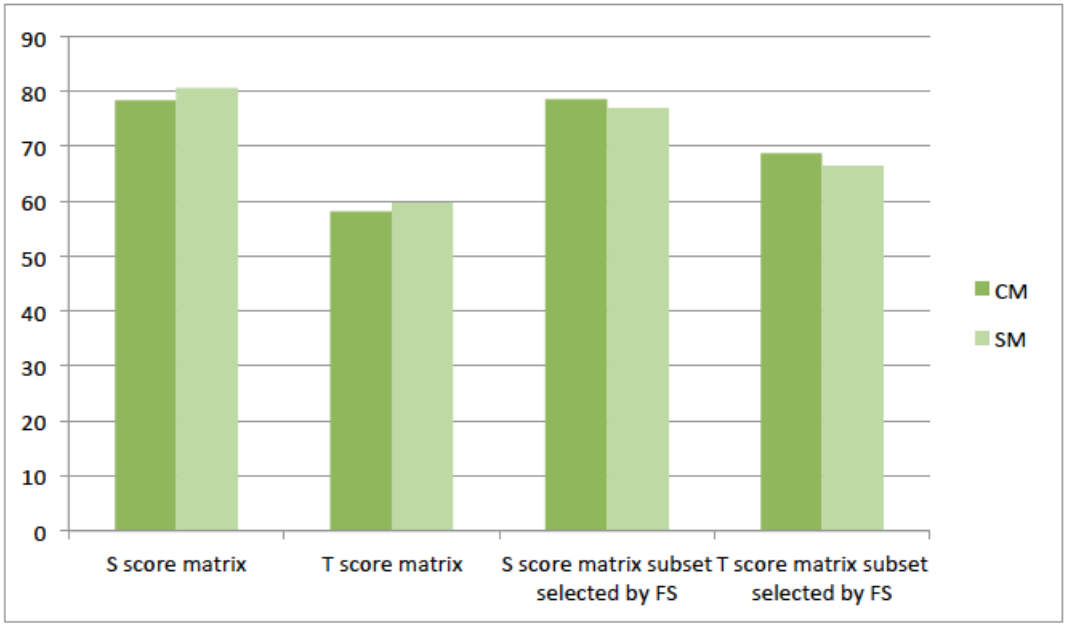

FIGURE 4.2.5: Comparison of prediction of CaM-binding proteins accuracies between classification results by 1-NN for matrixes with SLiMs obtained from SM and matrixes with SLiMs obtained from CM.

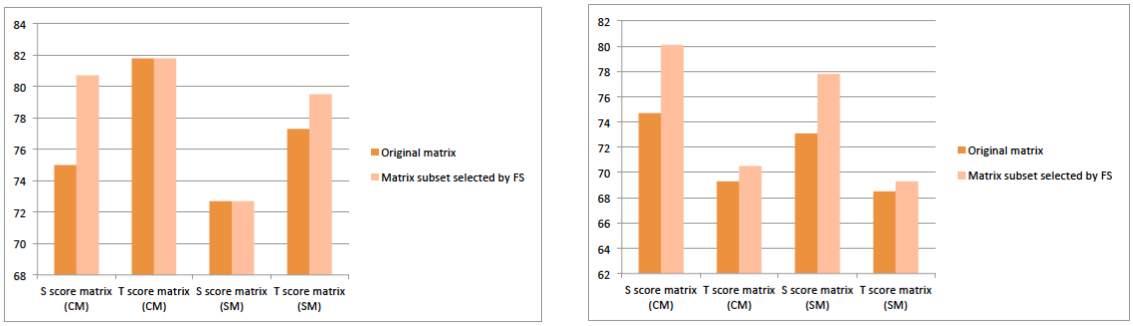

FIGURE 4.2.6: Comparison of accuracies of classification results on PPIs by 1-NN (left) and Random Forest (right) for original matrices obtained from SM and CM, with the results for matrices after feature selection.

Observing Figure 4.2.6 we find that, only the results with FS are equal to the results without FS for the results on the T score matrix with CM and the S score matrix with SM. Other classification all perform better with FS than non-FS using either 1-NN or RF for prediction of PPIs.

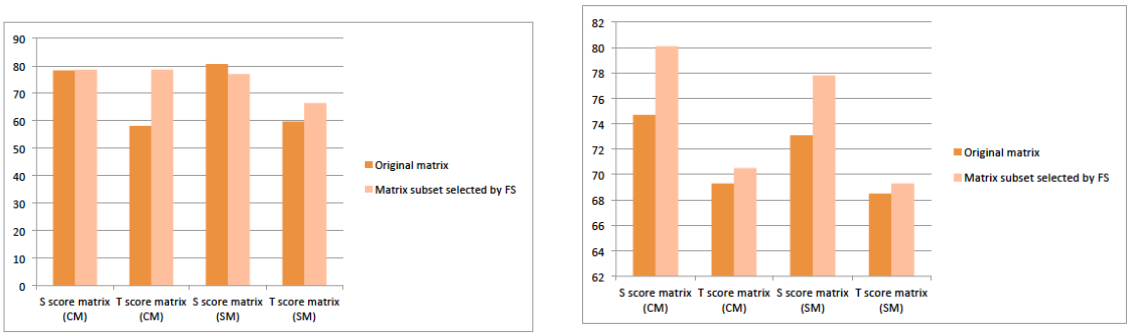

FIGURE 4.2.7: Comparison of accuracies of classification results on CaM-binding proteins by 1-NN (left) and Random Forest (right) for original matrixes obtained from SM and CM, with the results for matrixes after feature selection.

Figure 4.2.7 indicates the non-FS VS FS results for CaM-binding proteins. When using the 1-NN classifier, classification + FS achive higher accuracies on the S score matrix using CM, the T score matrix using CM and T score matrix with SM. Only on the S score matrix using SM, the non-FS obtained better results than FS. When using the RF classifier, classification + FS achieves better results on all score matrices.

CHAPTER 5

Conclusion and Future Work

5.1 Contributions

We propose one method with five different variances for prediction of high-throughput protein-protein interactions and prediction of Calmodulin Binding proteins using short linear motifs. Our method shows promising results and demonstrates that information contained in SLiMs is highly relevant for accurate prediction of high-throughput PPIs and CaM-binding proteins. The Sliding Window Scoring method is useful for scoring the sites and obtaining the datasets for classification.

As for prediction of PPIs, most of the classifiers perform better on the scores divided by the number of SLiMs. The classification experiments yield good results on the datasets with SLiMs obtained from both of the CM and SM approaches. The classification experiments yielded 86.4% accuracy when using SVM-Polynomial classifier on the scores divided by the number of the SLiMs with the SLiMs yielded from the CM dataset, which is the highest accuracy among all of the experimented results in this research. Our results also show that feature selection is necessary when using SVM-Polynominal, Random Forest, 1-NN and Decision Tree classifiers for these datasets.

For predicition of CaM-binding proteins, the classification experiments yield good results on the datasets with SLiMs obtained from both of the SM and CM approaches. The classification experiments obtained 80.6% accuracy when using 1-NN as a classifier on the total scores obtained from SM, which is the highest accuracy among all of the experiments.

Moreover, feature selection plays a key role in classification process on both prediction of PPIs and CaM-binding proteins. Most classifiers perform better after feature selection.

5.2 Future Work

Possible extensions of this work include investigating the SWS method on prediction of other types of protein-protein interactions. Also, possible extension to this work is to investigate the motifs obtained by MEME and relate them to existing families of calcium-binding motifs, possibly discovering new motifs of families. Finally, another extension to this work is to combine structural and SLiM data in order to provide a better insight of the location of the motifs on the interface, role on the interaction and other aspects.

REFERENCES

[1] (2014). https://sourceforge.net/projects/weka/files/weka-3-7/3.7.11/.

[2] Ahmed, H. R. and Glasgow, J. I. J. (2014). Pattern discovery in protein networks reveals high-confidence predictions of novel interactions. *AAAI*, pages 2938–2945.

[3] Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., Ren, J., Li, W. W., and Noble, W. S. (2009). Meme suite: tools for motif discovery and searching. *Nucleic Acids Research*, page gkp335.

[4] Bailey, T. L. and Elkan, C. J. (1995). The value of prior knowledge in discovering motifs with meme. *Ismb*, 3:21–29.

[5] Bailey, T. L. and Elkan, M. B. C. J. (2010). The value of position-specific priors in motif discovery using meme. *BMC Bioinformatics*, 11(1):1.

[6] Bailey, T. L., Williams, N., Misleh, C., and Li, W. (2006). Meme: discovering and analyzing dna and protein sequence motifs. *Nucleic Acids Research*, pages W369–W373.

[7] Blohm, P., Frishman, G., Smialowski, P., Goebels, F., Wachinger, B., Ruepp, A., and Frishman, D. R. (2014). Negatome 2.0: a database of non-interacting proteins derived by literature mining, manual annotation, and protein structure analysis. *Nucleic Acids Research*, page gkt1079.

[8] Bork, P., Jensen, L. J., von. Mering, C., Ramani, A. K., Lee, I., and Marcotte, E. M. (2004). Protein interaction networks from yeast to human. *Current Opinion in Structural Biology*, 14(3):292–299.

[9] Carbonell, Jaime, G., Michalski, R. S., and Mitchell, T. M. (1983). An overview of machine learning. *Machine learning*, pages 3–23.

[10] Chang, C. C. and Lin, C. J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(23):27.

[11] Cock, P. J., Antao, T., Chang, J. T., Chapman, A. B., Cox, C. J., and Dalke, A. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423.

[12] Davey, N. E., Haslam, N. J., Shields, D. C., and Edwards, R. J. (2010). Slimfinder: a web server to find novel, significantly over-represented, short protein motifs. *Nucleic Acids Research*, page gkq440.

[13] Davey, N. E., Haslam, N. J., Shields, D. C., and Edwards., R. J. (2011). Slimsearch 2.0: biological context for short linear motifs in proteins. *Nucleic Acids Research*, 39(2):W56–W60.

[14] Frank, E., Hall, M., Trigg, L., Holmes, G., and Witten, I. H. (2004). Data mining in bioinformatics using weka. *Bioinformatics*, 20(15):2479–2481.

[15] Gunn, S. R. (1998). Support vector machines for classification and regression. *ISIS technical report*, page 14.

[16] Guo, Y., Yu, L., Wen, Z., and Li, M. (2008). Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *Nucleic Acids Research*, 36(9):3025–3030.

[17] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations*, 11(1).

[18] Haslam, J., N., Shields, and C., D. (2012). Profile-based short linear protein motif discovery. *Bmc Bioinformatics*, 13(1):1.

[19] Krogan, N. J., Cagney, G., Yu, H., Zhong, G., Guo, X., Ignatchenko, A., and Punna, T. J. (2006). Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440(7084):637–643.

[20] Lei, C. and Ruan, J. (2013). A novel link prediction algorithm for reconstructing protein?protein interaction networks by topological similarity. *Bioinformatics*, 29(3):355–364.

[21] Li, Y., Rezaei, B., Ngom, A., and Rueda, L. (2015). Prediction of high-throughput protein-protein interactions based on protein sequence information. *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–6.

[22] McAndrew, A. (2004). An introduction to digital image processing with matlab notes for scm2511 image processing. *School of Computer Science and Mathematics, Victoria University of Technology*, pages 1–264.

[23] Mintseris, J. and Weng, Z. (2005). Structure, function, and evolution of transient and obligate protein-protein interactions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(31):10930.

[24] Nooren, I. and Thornton, J. (2003). Diversity of protein-protein interactions. *The EMBO Journal*, 22(14):3486–3492.

[25] Park, S. H., Reyes, J. A., Gilbert, D. R., Kim, J. W., and Kim, S. J. (2009). Prediction of protein-protein interaction types using association rule based classification. *BMC Bioinformatics*, 10(1):1.

[26] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., and Vanderplas, J. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

[27] Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.

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

[29] Rueda, L. and Pandit, M. (2014). A model based on minimotifs for classification of stable protein-protein complexes. *Computational Intelligence in Bioinformatics and Computational Biology, 2014 IEEE Conference*, pages 1–6.

[30] Saeys, Y., Inza, I., and Larraaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

[31] Safavian, S. R. and Landgrebe, D. (1990). A survey of decision tree classifier methodology.

[32] Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS One*, 10(3):e0118432.

[33] Schiller, M. R., Mi, T., Merlin, J. C., Deverasetty, S., Gryk, M. R., Bill, T. J., and Brooks, A. W. (2011). Minimotif miner 3.0: database expansion and significantly improved reduction of false-positive predictions from consensus sequences. *Nucleic Acids Research*, page gkr1189.

[34] Sharma, T. C. and Jain, M. (2013). Weka approach for comparative study of classification algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(4):1925–1931.

[35] Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., and Jiang, H. (2007). Predicting protein-protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences*, 104(11):4337–4341.

[36] Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F. H., Goehler, H., and Timm, J. (2005). A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122(6):957–968.

[37] Stevens and C., F. (1983). Calmodulin: an introduction. *Canadian Journal of Biochemistry and Cell Biology*, 61(8):906–910.

[38] Wikipedia (2016a). Fasta format.

[39] Wikipedia (2016b). Matthews correlation coefficient.

[40] Wikipedia (2016c). Multilayer perceptron.

[41] Wikipedia (2016d). Polynomial kernel.

[42] Wikipedia (2016e). Sequence motif.

[43] Yap, K. L., Kim, J., Truong, K., Sherman, M., Yuan, T., and Ikura, M. (2000). Calmodulin target database. *Journal of Structural and Functional Genomics*, 1(1):8–14.

[44] Yu, H., Braun, P., Yldrm, M. A., Lemmens, I., Venkatesan, K., Sahalie, J., and Hao, T. J. (2008). High-quality binary protein interaction map of the yeast interactome network. *Science*, 322(5898):104–110.

[45] Zhou, Y., Zhou, Y. S., He, F., Song, J., and Zhang, Z. (2012). Can simple codon pair usage predict protein?protein interaction? *Molecular BioSystems*, 8(5):1396–1404.

[46] Zhu, H., Domingues, F., Sommer, I., and Lengauer, T. (2006). Noxclass: prediction of protein-protein interaction types. *BMC Bioinformatics*, 7(1):1.

VITA AUCTORIS

NAME:	Yixun Li
PLACE OF BIRTH:	Bengbu, Anhui province, China
EDUCATION:	Tianjin Polytechnic University, B.Eng., Software Engineering, Tianjin, China, 2014
	University of Windsor, M.Sc in Computer Science, Windsor, Ontario, 2016