COEVOLUTION BASED PREDICTION OF PROTEIN-PROTEIN INTERACTIONS WITH REDUCED TRAINING DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BAHAR PAMUK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

FEBRUARY 2009

Approval of the thesis:

COEVOLUTION BASED PREDICTION OF PROTEIN-PROTEIN INTERACTIONS WITH REDUCED TRAINING DATA

submitted by **BAHAR PAMUK** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Müslim Bozyiğit
Head of Department, **Computer Engineering** _____

Assist. Prof. Dr. Tolga CAN
Supervisor, **Computer Engineering** _____

Examining Committee Members:

Asst. Prof. Dr. Tolga Can
METU, CENG _____

Prof.Dr. Volkan Atalay
METU, CENG _____

Assoc. Prof. Ferda Nur Alpaslan
METU, CENG _____

Prof.Dr. Göktürk Üçoluk
METU, CENG _____

Asst. Prof. Dr. Çiğdem Gündüz Demir
Bilkent, CS _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: BAHAR PAMUK

Signature :

ABSTRACT

COEVOLUTION BASED PREDICTION OF PROTEIN-PROTEIN INTERACTIONS WITH REDUCED TRAINING DATA

Pamuk, Bahar

M.S., Department of Computer Engineering

Supervisor : Assist. Prof. Dr. Tolga CAN

February 2009, 60 pages

Protein-protein interactions are important for the prediction of protein functions since two interacting proteins usually have similar functions in a cell. Available protein interaction networks are incomplete; but, they can be used to predict new interactions in a supervised learning framework. However, in the case that the known protein network includes large number of protein pairs, the training time of the machine learning algorithm becomes quite long. In this thesis work, our aim is to predict protein-protein interactions with a known portion of the interaction network. We used Support Vector Machines (SVM) as the machine learning algoritm and used the already known protein pairs in the network. We chose to use phylogenetic profiles of proteins to form the feature vectors required for the learner since the similarity of two proteins in evolution gives a reasonable rating about whether the two proteins interact or not. For large data sets, the training time of SVM becomes quite long, therefore we reduced the data size in a sensible way while we keep approximately the same prediction accuracy.

We applied a number of clustering techniques to extract the most representative data and features in a two categorical framework. Knowing that the training data set is a two dimensional

matrix, we applied data reduction methods in both dimensions, i.e., both in data size and in feature vector size. We observed that the data clustered by the k-means clustering technique gave superior results in prediction accuracies compared to another data clustering algorithm which was also developed for reducing data size for SVM training. Still the true positive and false positive rates (TPR-FPR) of the training data sets constructed by the two clustering methods did not give satisfying results about which method outperforms the other. On the other hand, we applied feature selection methods on the feature vectors of training data by selecting the most representative features in biological and in statistical meaning. We used phylogenetic tree of organisms to identify the organisms which are evolutionarily significant. Additionally we applied Fisher's test method to select the features which are most representative statistically. The accuracy and TPR-FPR values obtained by feature selection methods could not provide to make a certain decision on the performance comparisons. However it can be mentioned that phylogenetic tree method resulted in acceptable prediction values when compared to Fisher's test.

Keywords: Phylogenetic profiles, Support Vector Machines, K-means clustering, Phylogenetic tree, Protein interaction networks

ÖZ

PROTEİN-PROTEİN ETKİLEŞİMLERİNİN KÜÇÜLTÜLMÜŞ ÖĞRENME VERİSİ İLE BİRLİKTE EVRİMLEŞMEYE DAYALI TAHMİNİ

Pamuk, Bahar
Yüksek Lisans, Bilgisayar Mühendisligi Bölümü
Tez Yöneticisi : Yard. Doç Tolga CAN

Şubat 2009, 60 sayfa

Bir hücre içerisinde aynı görevi gerçekleştiren proteinler çoğunlukla birbirleriyle etkileştikleri için protein-protein etkileşim ağları proteinlerin fonksiyonlarının bulunmasında önemli rol oynarlar. Protein çiftlerinin bir kısmı bilinen bir protein etkileşim ağında, henüz belirlenmemiş protein çiftleri makina ögrenme algoritmaları vasıtasıyla bilinen kısım kullanılarak bulunabilir. Ancak protein ağlarının çok sayıda protein çifti içerdigi bir durumda makina ögrenme algoritmasının ögrenme süresi oldukça uzun olacaktir. Bu tez çalışmasında etkileşimlerinin bir kısmının bilindiği bir etkileşim ağının bilinmeyen kısmını bulmayı deneyler yoluyla gerçekleştirmeyi amaçladik. Makina öğrenme algoritması olarak Destek Vektör Makinalari (DVM)'nı ve bir ağ içerisinde bilinen protein çiftlerini kullandik. Evrimsel açıdan iki proteinin birbirine yakın olması, bu iki proteinin etkileşimleri hakkında iyi bir değerlendirme vereceği için, öğrenici için gerekli olan öznitelik vektörü olarak proteinlerin filogenetik profillerini kullandık. Büyük boyuttaki veriler için Destek Vektor Makinalarının ögrenme süreleri uzun olacağindan veriyi doğruluk oranlarını koruyarak makul bir şekilde küçülttük.

İki kategorili bir çati altında veriyi küçültmek amacıyla en sembolik veriyi seçmek için bazı kümeleme tekniklerini uyguladık. Verinin iki boyutlu bir matris olduğunu göz önünde bu-

lundurarak, veri küçültme metotlarını iki boyutta da uyguladık (hem verinin boyutunda hem öznitelik vektörünün boyutunda). K-means tekniği ile kümelenen veri kümelerinin tahmin doğruluklarında veriyi SVM ögrenmesi için küçülten başka bir kümeleme algoritmasına kıyasla daha üstün sonuçlar verdiğini gözlemledik. Yine de iki algoritma tarafından da oluşturulan öğrenme verisinin TPR-FPR değerleri, hangi metodun daha üstün olduğu konusunda tatmin edici sonuçlar vermedi. Diğer yandan, ögrenme verilerinin özellik vektörleri üzerinde biyolojik ya da istatistiksel anlamda en sembolik özellikleri seçmek için özellik seçme metotlarini uyguladık. Evrimsel olarak en önemli olan organizmaları belirlemek için organizmaların filogenetik ağaçlarını kullandık. Ayrıca, istatistiksel olarak en sembolik özellikleri seçmek için Fisher's test metodunu uyguladık. Özellik seçme metotlarından elde edilen doğruluk ve TPR-FPR değerleri performans kıyaslaması yapmak konusunda kesin bir ayrım yapmayi saglayamadı. Yine de, filogenetik ağaç metodunun Fisher's test ile kiyaslandığında kabul edilebilir tahmin değerleri verdiği söylenebilir.

Anahtar Kelimeler: Filogenetik profiller, Destek Vektör Makinaları, K-means kümeleme, Filogenetik ağaçlar, Protein etkileşim ağları

To my family

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Asst. Prof. Dr. Tolga CAN for his enthusiasm, positivity towards every misadventure, sharings of his expertise in his field, all the technical and spiritual support and suggestions he provided and his sensibility.

I wish to thank to my dear friend Hande Çelikkanat, being my companion, for the confidence she provides at any moment, for all the favors she never withholds and for all the valuable thoughts we shared.

Even though we meet seldomly, for making me feel all her pure sentiments and for the peace she conveyed with her tranquility, I am thankful to Çağla Okutan.

For the invaluable companionship she provided, for the unforgettable foods (including burned ones) we cooked, for her endless energy, I want to thank to my companion Sevgi Yaşar.

I wish to thank to Gülşah Karaduman, my dear friend and neighbor, for the good fellowship she shows at all times.

For providing the confidence of knowing to be supported and understood and for his everlasting encouragements during thesis study I wish to thank to İbrahim Taşyurt.

I deeply thank to Burçin Sapaz for his handy advices, favorable thoughts and for supporting by bracing me up during the thesis work.

Thanks to my office mate, Gencay Evirgen for his generosity, everlasting cheerfulness and prayers for me.

Thanks to the admins of HPC cluster ”nar” in our department, Ahmet Ketenci and Çelebi Kocair for their understanding and support while running our experiments.

This work is supported by TÜBITAK 2210 scholarship program.

We also thank to Alper Söyler for providing phylogenetic profiles for the yeast organism.

Thanks to Burçin Selçuk for being so close to me as a sister even from a great distance.

I would like to state my gratitude to my family for their encouragements and the loving environment that they provided to me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Definition and Motivation

Protein-protein interaction is an important aspect in systems biology. A systems level understanding of the signaling pathways and molecular complexes in a cell provides more accurate identification of cellular functions of proteins, better understanding of biological and pathological processes, and more confident drug target identification. With the help of a protein-protein interaction (PPI) network, one can select molecular compounds which specifically disrupt certain protein-protein interactions that is related to the disease pathway.

In recent years, protein-protein interaction datasets for an increasing number of organisms have been made publicly available with the help of high-throughput screening techniques. Main experimental techniques for discovering protein-protein interactions are the yeast two-hybrid (Y2H) and affinity purification with mass spectrometry (APMS). However, it is known that these experimental techniques have high false-positive and false-negative rates; therefore, in addition to these experimental techniques, computational techniques that use additional biological information such as co-expression, co-localization, and co-evolution, have been developed. The main challenge of a genome-wide prediction of protein-protein interactions is that the protein pairs that do not interact outnumber interacting proteins significantly. For example, there are 100,000 estimated interactions out of 18 million possible in the yeast organism. For a newly sequenced organism, the challenge is even bigger that little amount of additional biological knowledge is available for such an organism. Therefore, it is important to develop accuracte PPI prediction techniques that use protein sequence information only.

In this thesis, our goal is to develop a machine learning based technique for prediction of

protein-protein interactions based on co-evolution. Co-evolution can be inferred from phylogenetic profiles which can be derived from protein sequence information alone. We generate phylogenetic profiles as high dimensional feature vectors by comparing each protein of the organism to proteins of a number of other fully sequenced genomes. Given the phylogenetic profiles of all the proteins of an organism, the problem can be stated as the classification of all possible protein pairs as interacting or non-interacting. This is a binary classification problem. We learn a discriminative model using Support Vector Machines to distinguish between these two classes. In that sense, we provide a supervised solution to this problem.

1.2 Related Work

1.2.1 Using Support Vector Machines (SVM) to infer Protein-Protein Interactions

There have been studies which use Support Vector Machines (SVM) to predict protein-protein interactions. The AC method proposed by Guo et al. [27] uses the neighborhood of amino acids in a protein sequence by means of Auto Covariance method to predict the protein-protein interactions. We used the same data set and the same method as they used to produce the training and test sets to evaluate our method. Another technique proposed by Bock et al. [26] uses the primary structure of proteins together with the physicochemical properties of a known database of protein interactions as training data for SVM to make predictions on protein-protein interactions. They used residue properties of amino acids such as charge, hydrophobicity and surface tension to construct feature vectors which is an independent knowledge from coevolution which we made use of in our method. There is also another method by Martin et al. [28] which solves the protein-protein interaction problem by training an SVM with product descriptions of protein pairs. They encode the variable length amino acids to signatures by using their neighbors. All of these studies use the physical or chemical properties of proteins while we consider the coevolution knowledge of proteins by using their phylogenetic profiles.

1.2.2 Using Phylogenetic Profiles to Infer Protein-Protein interactions

In the study conducted to express the significance of phylogenetic profiles for discovering the functional linkages among proteins by Juan et al. [1] it is inferred that proteins having similar phylogenetic profiles are functionally linked assuming it is likely that proteins in the same metabolic pathway or cellular system are co-inherited during evolution. The idea of coevolution using phylogenetic profiles has been used by many researchers in the prediction of protein-protein interactions. Pellegrini et al. [24] demonstrates the value of phylogenetic profiles of proteins in detecting their functions by simply comparing the phylogenetic profiles and counting the numbers of bits changed which is a basic way to calculating the similarity between two profiles. Wu et al. [23] uses the similarity of phylogenetic profiles by applying a method to relax the restrictions that phylogenetic profiles require by a biological pressure measure. They use different correlation measures between two vectors. Bowers et al. [13] compute the probability of coevolution based on hypergeometric distribution. In other words, given two phylogenetic profiles they convert it into a probability value that represents their confidence on their coevolution. They use this probability value in an integrative framework to derive functional association of proteins. Kim and Subramaniam [14] use a mutual information function based on the Shannon entropy to indicate the level of similarity between two phylogenetic profiles. Vert [15] developed a tree kernel which provided a better similarity measure between two phylogenetic profiles. He used this kernel to predict the functional class of a gene. Sato et al. [17] improve Pearson's correlation coefficient by proposing partial correlation coefficient as a function of similarity between two profiles. Juan et al. [18] analyze the network of profile similarities to account for groups of coevolving proteins and reduce the noise associated with various factors that make-up a phylogenetic profile. Gonzales et al. [22] include the phenotype knowledge to phylogenetic profiles in order to extend the binary strings to continuous phenotypes and develop scoring functions to use them in pairs . All of these studies focus on providing a similarity measure that best captures the amount of co-evolution between two proteins. However, in this thesis, instead of using an explicit similarity function, we propose a machine learning approach which learns such a function implicitly on a training dataset.

Apart from defining a similarity function for phylogenetic profiles, there are some studies which try to refine the organisms selected for phylogenetic profiling. Sun et al. [19, 20] pro-

pose a phylogenetic approach to select representative organisms to construct a phylegenetic profile. They, then, apply existing similarity measures on the reduced phylgenetic profiles. We adopt their approach in this thesis. However, we do not use an explicit similarity function as mentioned above, and retain the reduced phylogenetic profiles as high dimensional vectors.

1.2.3 Reducing Size of Training Data for Support Vector Machines

There have been studies to reduce the size of the training data set for supervised learning with Support Vector Machines. Cervantes et al. [6] propose the ball-clustering technique to select representative data points for SVM training. Their approach is based on the number of support vectors in the original training set; therefore, reduces the number of training data points to a fixed number. In our approach, we use k-means clustering, in which, the user controls the number of representative data points by varying k.

1.3 Contributions

Our contribution in this thesis are threefold.

1) By employing a machine learning framework we avoid using similarity functions to indicate the level of co-evolution between two phylogenetic profiles. Previous studis focus on developing biologically accurate functions to infer the level of co-evolution between two phylogenetic profiles. However, we retain phylogenetic profiles as high dimensional vectors and the Support Vector Machine approach implicitly learns a discriminative function between pairs of phylogenetic profiles.

2) We propose a clustering based technique to reduce the number of training protein pairs. Compared to a previous techniqe, our method provides better accuracy when the number of selected training proteins pairs are equal.

3) We propose a biologically inspired feature selection technique which outperforms a widely adopted statistical feature selection technique. Our technique utilizes domain knowledge and makes use of the fact the each feature dimension corresponds to an organism. By using a phylogenetic tree of feature organisms to denote the relationships between them, we are able to select a better representative subset of organisms.

1.4 Thesis Outline

This thesis is organized as follows. In Chapter 2, we give the necessary background knowledge to understand the problem domain and the solutions we provide. In Chapter 3, we describe the datasets we have used and describe the technical details of the methods we propose. In Chapter 4, we give experimental results which demonstrate the utility of the proposed methods. In Chapter 5, we conclude the thesis with a summary and future directions.

CHAPTER 2

BACKGROUND

2.1 Proteins

Proteins are organic compounds which are constructed from amino acids and are responsible for numerous functions in a living cell. In a protein there are about 200-300 amino acids which are arranged in a linear chain and joined by peptide bonds. A peptide bond is formed when two molecules react with particular groups of each other and release H_2O. Proteins function via their three dimensional structures. The properties of proteins such as their structures, their physiochemical properties, locations in the living cell and their relationships with each other determine their functions and interactions with each other.

2.2 Amino Acid Sequences

The amino acid sequence of a protein is a string composed of the letters each representing one of the 20 different kinds of amino acids. An amino acid sequence characterizes the arrangement of amino acids in a protein and the structure of a protein. Also, the function of a protein can be determined by making use of the arrangement of amino acid sequences. The functional relationship between two proteins can be observed by making an alignment between their amino acid sequences. A sequence alignment which gives scores about the similarity of two proteins might give a rating about their functional closeness or whether the two proteins can be homologous or not.

Amino acid sequences can be aligned in pairwise or in multiple. There are some methods for the alignment of sequenes which are local alignment where only some portions in a sequence

are used to score and global alignment where the whole sequence is used for the alignments. There are online tools, which are the implementations of various alignment methods, available for amino acid sequence alignments. For pairwise sequence alignment FASTA [1] or BLAST [2] [29] can be used. For multiple alignment CLUSTALW [3] [31], TCOFFEE [4] [32] or Muscle [5] [33] can be applied.

2.3 Protein-Protein Interaction Networks

Protein interactions are essential for making predictions of functions of proteins. Protein interactions are observed during signal transduction (i.e. the signals outside the cell are transferred inside the cell), generating a protein complex or modifying a protein.

Protein-protein interaction networks are graphs that represent the interaction involvement of protein pairs. In a protein interaction network the protein pairs that are connected by an edge are perceived as interacting pairs and rest are the noninteracting ones. Below is a sample interaction graph [6] of yeast organism.

In a protein interaction network, the edges between the proteins might include weights where these weights can be the functional correlation between the proteins or the level of confidence assigned to that interaction [7].

There are numerous protein-protein interaction databases which provide the protein interaction data of various organisms and are mostly constituted by hand-made experiments done by experts. The three protein interaction databases that we benefited from are:

- MIPS (Munich Information Center for Protein Sequences) Mammalian Protein-Protein Interaction Database [7] [36] which includes the physical interactions of proteins that are determined only by hand made experiments since it is the most reliable way to extract the interaction knowledge.

[1] http://www.ebi.ac.uk/Tools/fasta33/index.html
[2] http://blast.ncbi.nlm.nih.gov/Blast.cgi
[3] http://www.ebi.ac.uk/Tools/clustalw2/index.html
[4] http://www.ch.embnet.org/software/TCoffee.html
[5] http://www.ebi.ac.uk/Tools/muscle/index.html
[6] http://www.math.cornell.edu/ durrett/RGD/RGD.html
[7] http://mips.gsf.de/proj/ppi/

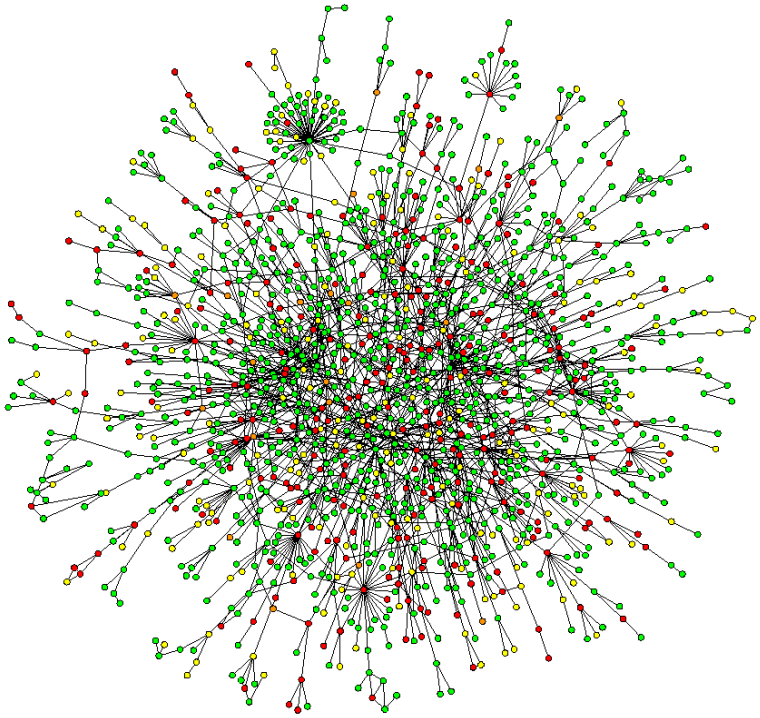

Figure 2.1: Protein Interaction Network of yeast organism

- DIP [8] [37] (Database of Interacting Proteins) which includes the protein-protein interactions discovered by both hand-made experiments and computational approaches.

2.4 Phylogenetic Profiles

Phylogenetic profile of a protein is a string constituted by 1s and 0s that encodes the presence or absence of homologs of a protein in other organisms. In construction of a phylogenetic profile of a protein, the homologs of the protein are searched against the other organisms. The search of homologoues of a protein can be done by aligning the sequence of that protein with proteins of others organisms.

The alignments can be done via any sequence alignment tool some of which are dictated in Section 2.2 and each alignment of sequences are scored by the tool. If the alignment score of a protein in a protein of another organism is above a predetermined cut-off value, it means the protein has a homologue in that organism and the correspondant value in the profile becomes

[8] http://dip.doe-mbi.ucla.edu/

1, otherwise there is no homologue of that protein and the value is set to 0 in the string.

	bqu	cfa	eba	sme	rxy	dpyo ...
YOR070C	0	1	0	0	0	1 ...
YAL067C	0	0	0	0	1	0 ...
YDR338C	0	1	1	1	0	1 ...

Figure 2.2: A portion of phylogenetic profiles of 3 proteins belonging to the yeast organism

In this thesis work, the phylogenetic profile data is constructed by calculating the sequence alignment score by the help of the BLAST tool. Figure 2.2 is a data portion of phylogenetic profiles of 3 proteins of Saccharomyces cerevisiae yeast organism with homology search against six other organisms that we used in our experiments.

2.5 Phylogenetic Trees

Phylogenetics is the study of evolutionary relatedness among various groups of organisms which is discovered through molecular sequencing data. Evolution is a branching process where populations alter by time, separate into branches or hybridize together or exposed to extinction. This evolution process is used to construct a full tree. Evidence from morphological, biochemical, and gene sequence data suggests that all organisms on Earth are genetically related, and the genealogical relationships of living things can be represented by a vast evolutionary tree, the Tree of Life [9] which represents the phylogeny of organisms.

In phylogenetic studies, the most suitable way to visualize the evolutionary relationships among a group of organisms is by phylogenetic trees. Figure 2.3 is a sample phylogenetic tree to present its components. In this figure, a **node** represents a taxonomic unit, i.e an existing species or an ancestor. They are usually referred to as Hypothetical Taxonomic Units (HTUs) since they are not directly observed. **Root** is the common ancestor of all taxa. A **branch** is an evolutionary relationship among taxonomic units. The **branch length** exhibits the number of changes that have occurred in the branch. That is to say it represents the evolutionary distance between taxonomic units. Hence, in a phylogenetic tree the species are located at the leaves of the tree. A phylogenetic tree is constructed by the usage of multiple sequence alignments whose scores represent the evolutionary distances. There are three

[9] http://tolweb.org/tree/learn/concepts/whatisphylogeny.html

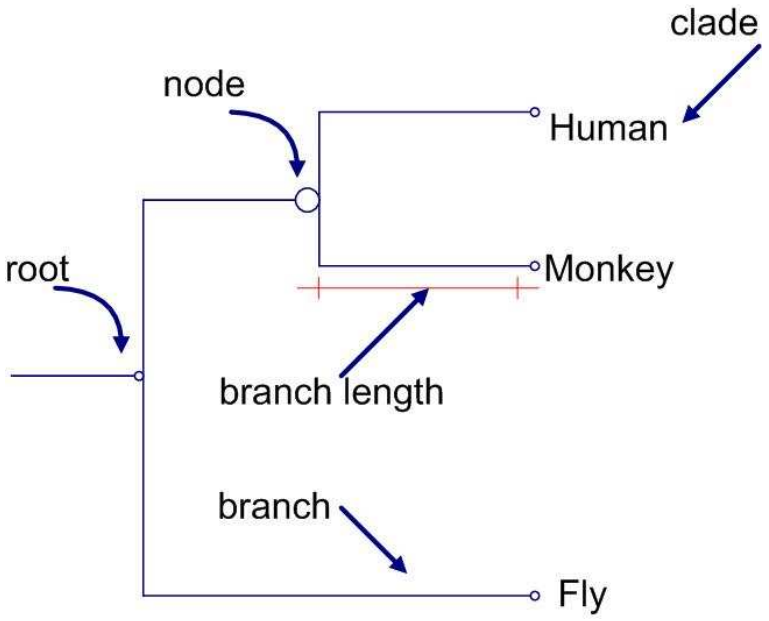

Figure 2.3: A sample phylogenetic tree with its components

main methods of constructing phylogenetic trees: Distance based methods such as Neighbour Joining [12], parsimony based methods such as Maximum Parsimony, and character based methods such as Maximum Likelihood or Bayesian Inference.

A **rooted** tree as in Figure 2.3, is a directed tree with a unique node corresponding to the most recent common ancestor of all the entities at the leaves of the tree. An **unrooted** tree illustrates the relatedness of the leaf nodes without making assumptions about common ancestry. Unrooted trees can be obtained by omitting the root of a rooted tree. The root of an unrooted tree can be obtained by various ways. The methods to construct a phylogenetic tree emphasized above, may end up with unrooted trees. The adjacent taxa may not be closely related to each other in an unrooted tree. To obtain a root for an unrooted tree, an outgroup which is known to be branched before all other nodes in the tree can be included to the tree.

2.6 Fisher's Exact Test

Fisher's exact test is a statistical significance test used in the analysis of categorical data. The test is usually used to examine the significance of the association between two variables in a two by two contingency table. In a binary decision problem, the decision made by the

classifer can be represented in a structure known as a confusion matrix or contingency table [9]. Contingency tables are used to analyze the relationship between two variables. A sample contingency table is given in Table 2.1:

Table 2.1: A sample contingency table

	B1	B2	Totals
A1	a	b	a+b
A2	c	d	c+d
Totals	a+c	b+d	n

The probability of obtaining those values in the table is calculated according to the following hypergeometric distribution:

$$p = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} = \frac{(a+b)!(c+d)!(a+c!)(b+d)!}{n!a!b!c!d!}$$

where n is $a + b + c + d$.

The p-value gives the exact probability of observing observing this particular arrangement of the data.

Chi-Square Test

An altenative method for testing a statistical hypothesis is to use the Chi-Square Test. Fisher's exact test is applied to data with two by two contingency tables whereas a chi-square test is used on tables with more rows and columns; i.e it is more suitable for the data with larger number of categories. A chi-square test is not suitable for the situations where the expected values in any of the cells of the contingency table is below 10.

Chi-square tests a null hypothesis that the frequency distribution of certain events observed in a sample is consistent with a particular theoretical distribution. A chi-square statistic value is calculated according to the below formula:

$$\mathcal{X}^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i},$$

where $\mathcal{X}^2$ is the test statistic value, O_i is the observed frequency of the i^{th} category and E_i is the theoretical frequency of the i^{th} category.

When analyzing contingency tables with two rows and two columns, either Fisher's exact test or the chi-square test could be chosen. The Fisher's test is the best choice as it always gives the exact P value. The chi-square test is simpler to calculate but yields only an approximate P value. If the numbers in the contingency table are very small, the chi-square test should be avoided. When the numbers are larger, the P values reported by the chi-square and Fisher's test will be very similar.

In our case the data includes two kinds of categories and the values in the contingency tables were more suitable for using Fisher's exact test. Thus, we preferred to use Fisher's exact test in feature vector selection which is described in detail in Chapter 3.

2.7 Data Classification

Data classification is the problem of detecting which class a data point belongs to when a set of points are given as belonging to a class. Data classification is an essential component in the scope of this thesis work. We preffered to apply Support Vector Machines as a data classification algorithm in machine learning context where the domain of data is given to the learner explicitly. Then we compared the results of Support Vector Machine with another machine learning algorithm, Bayesian learning, where the reduction of feature vectors to a scalar causes some data loss, because the data can not be fed to the Bayesian learner completely.

2.7.1 Support Vector Machines

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. In SVMs the data points are represented as $p-$dimensional vectors where each data point belongs to one class. SVM maps the input vectors into a high dimensional feature space by means of some non-linear mapping function [5]. The trick is to find the $p-1$ dimensional hyperplane which seperates the data points with maximum margin. Meaning that, the hyperplane which maximizes the distance between the nearest point to the hyperplane is chosen as the **maximum-margin hyperplane**. In the data classification problem dictated in this thesis work, we have 2 classes to seperate.

We have a set of points S with two classes of data. The dot product between two vectors are re-

quired for the linear classifiers where the data set is binary labelled. $S = \{(x_i, c_i)|x_i \epsilon \mathbb{R}^p, c_i \epsilon \{-1, 1\}\}_{i=1}^n$ where c_i represents which class the data belongs to, i.e. whether +1 or -1. Each x_i is a $p-$ dimensional vector. The maximum margin hyperplane we are looking for is represented as the set of points x satisfying:

$$w \cdot x - b = 0 \tag{2.1}$$

So the equations can be rewritten as $c_i(w \cdot x_i - b) \geq 1$, where $1 \leq i \leq n$.
where w and b should be chosen to minimize $\|w\|$.

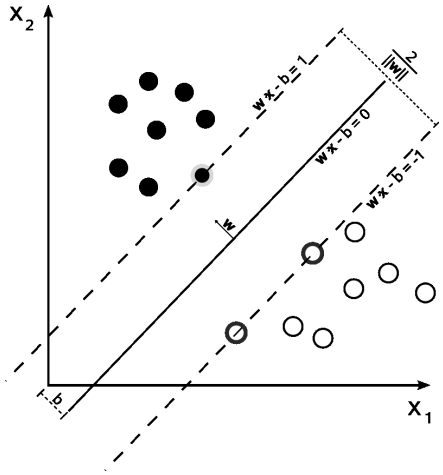

Figure 2.4: Maximum margin hyperplane with two classes[8]

The data on the margins in Figure 2.4 are called the **support vectors**. The distance to maximize on the hyperplane is represented as a quadratic problem [5]:

$$\rho(w, b) = \frac{2}{|w|} = \frac{2}{\sqrt{w \cdot w}} \tag{2.2}$$

The transformation of the input vectors in n dimensions into p dimensions is done via a p dimensional function: $\phi \mathbb{R}^n \rightarrow \mathbb{R}^p$ $\phi(x_i) = \phi_1(x_i), \phi_2()x_i, ..., \phi_p(x_i)$

It is shown in Cortes et al. [5] that the vector w can be written by the linear combination of training vectors as:

$$w = \sum_{i=1}^{l} y_i \alpha_i^0 x_i \tag{2.3}$$

where x_i are the support vectors and y_i is the label of the i^{th} feature vector(either 1 or -1).

To classify an unknown vector x , the vector is transformed into the feature space and then the sign of the below function is taken:

$$f(x) = w \cdot \phi(x) + b = \sum_{i=1}^{l} y_i \alpha_i \phi(x) \cdot \phi(x_i) + b \tag{2.4}$$

The mapping of points into a Hilbert space (a vector space closed under dot products) is achieved by the kernel functions which matches the point pairs to their dot products in Hilber space. A kernel function must be continuous, symmetric, and have a positive definite gram matrix[10]. If the classifier is linear then the kernel functions is $K(x_i, x_j) = x_i^T x_j$. Otherwise the points are transformed to a higher dimensional space by $\Phi : x \rightarrow \phi(x)$ and the kernel function is $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$.

The basic four types of kernels functions are as follows [SVM]:

- linear: $K(x_i, x_j) = x_i^T x_j$
- polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- radial basis function (RBF): $K(x_i, x_j) = exp(-\gamma ||x_i - x_j||^2), \gamma > 0$
- sigmoid: $K(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$.

2.7.1.1 Grid Search in SVMs

In SVM, C and γ parameters controls the trade-off between training error and generalization ability [25]. In the RBF kernel, it is ciritical to choose the C and γ parameters of SVM to extract the optimum model. A good selection of C and γ pair leads to a good prediction performance. Once the C parameter is selected high, the margin becomes softer and the number of support vectors increase which may cause over-fitting of the data. It is better to choose a small C parameter as much as possible.

Cross-validation technique is made use for making experiments in order to select best parameter values in RBF kernel. In v-fold cross-validation, the data set is seperated into v parts equally where one part is used for testing and the rest $v - 1$ parts are used for training. After

[10] http://nlp.stanford.edu/IR-book/html/htmledition/nonlinear-svms-1.html

the data is seperated, the training and prediction processes are applied for v times on the data in order to be sure that all $1/v$ portion of the data is included in the test set. The aim of $v-$fold cross validation is to prevent the learner from overfitting.

The (C, γ) pairs are tried on the data set and after the $v-$ fold cross validation experiments done, the one with the best accuracy on avarage of the v experiments is picked. The search is done with exponentially growing values. The start and end points with the step number for incrementing the values are the inputs that are given to the grid search tool. For instance, below call for the grid script included in the libsvm-2.84[11] package, which is a commonly used SVM package, tries the pairwise combinations of C and γ values for `start` and `end` values of -1 and 2 respectively for C parameter with 1 as the inceremental step (the number to increment to reach the `end` from `start`). Likewise it uses the `start` and `end` values of 1 and 5 for γ parameter with 2 as the incremental step.

```
python grid.py -log2c -1,2,1 -log2g 1,5,2 dataset
```

2.8 K-means Clustering

K-means is a well-known data mining and unsupervised learning algorithm to classify or to group objects based on attributes/features into k number of group where k is a positive integer. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Thus the purpose of k-means clustering is to classify the data.

K-means clusters n objects into k clusters where $k < n$ in p dimensional vectorial space. The aim in this algorithm is to minimize the intra-cluster distances, i.e minimize the squared error function given by:
$V = \sum_{i=1}^{k} \sum_{x_j \epsilon S_i} (x_j - \mu_i)^2$
where x_j is a chosen point in i_{th} cluster and μ_i is the cluster center point of the $i_{th} cluster$

The algorithm works as follows:

- Begin with a decision on the value of k = number of clusters

[11] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

- k cluster center points are randomly assigned among the n data points; $\mu_1, \mu_2, ..., \mu_k$
- Until there are no changes in the cluster centers, i.e no changes in the assignment of data points
  - The data points are assigned to the cluster whose centroid is the nearest.
  - Update the centroid of each cluster. The centroid of a cluster is the average of all points in that cluster, i.e the arithmetic mean of all dimensions

Sometimes it can happen that the data set which is closest to a cluster center μ_i is empty. It is ensured that no cluster pairs have common elements in this algorithm. Also, the number of clusters, k in the algorithm, effects the results of the clustering so it should be carefully chosen. The optimal number of clusters for a data set can not be determined beforehand. One way is to run the algorithm for a number of times for different k values and choose the one which gives best results.

2.9 ROC (Receiver Operating Characteristic) Curves

For the performance of the methods that we experimented in the context of this thesis work, we plotted the ROC Curves to interpret the results of the experiments. A ROC curve, is a graphical plot of the sensitivity (True Positive Rate (TPR)) vs. 1 - specificity (False Positive Rate (FPR)) for a binary classifier system as its discrimination threshold is varied.

	Actual Pos.	Actual Neg.
Predicted Pos.	TP	FP
Predicted Neg.	FN	TN

According to the parameters in a confusion matrix of a classifier, the definitions of metrics used in ROC curves are:

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}$$

A prediction with perfect seperation of data has a ROC plot that passes through the upper left corner (100% sensitivity, 100% specificity) as in Figure 2.5. Therefore, the closer the ROC

plot is to the upper left corner, the higher the accuracy of the test. The ROC curve in Figure 2.5 can be a sample for the output of a method with rather high accuracy performance due to its tendency to upper left corner in the graph.

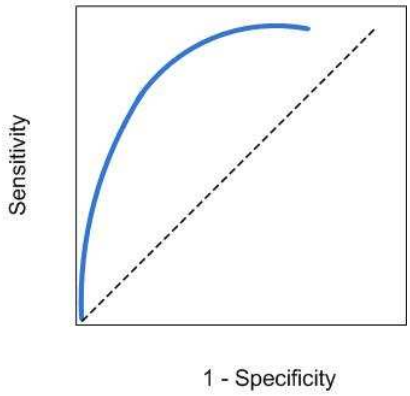

Figure 2.5: A sample ROC curve

CHAPTER 3

MATERIALS AND METHODS

3.1 Data Sets

Below we present the data sources and tools that we used to measure the performance of the data reduction methods which we applied on the data sets.

3.1.1 Phylogenetic Profiles

The phylogenetic profiles of proteins of Saccharomyces cerevisiae (yeast) organism was constructed via the BLAST tool as described in Chapter 2 over 450 organisms. The phylogenetic profile information of a protein is used as the feature vector belonging to that protein.The value ”1” was assigned to the corresponding index in the profile if the e-value returned from the BLAST tool is below 0.001, otherwise ”0” is given, so the feature vectors are composed of binary values. The shortened names of organisms used in construction of the phylogenetic profiles are listed in Appendix A.1.

3.1.2 Phylogenetic Tree Data

For feature vector selection, we used the phylogenetic tree data available in the Kyoto Encyclopedia of Genes and Genomes (KEGG) [1] database since the phylogenetic tree data in KEGG source includes all of the 450 organisms which were employed to generate the phylogenetic profiles. The phylogenetic tree in the KEGG database was the most adequate one among the other sources since it includes the most comprehensive data on account of comprising all the

[1] http://www.genome.jp/kegg/

organisms that we used.

In KEGG database, the computer representation of various data sources are available together with some tools that provide making search on the data sources, making analysis and drawing structures. The phylogenetic tree data provided in the KEGG database is not in a format that can be easily parsed (i.e. any structure to make the extraction of the tree simpler is not available), we directly used the taxonomy file in the ftp site provided[2]. This file includes the categorization of organisms at each depth of the tree which are represented by the lines starting by a '#' character. As usual, the organisms take place at the leaves with lines starting with the PIR-PSD (International Protein Sequence Database) [3] id. PIR-PSD is the database including the classified and functionally annotated protein sequences.

250 organisms selected by the phylogenetic tree method are represented in Appendix A.1.1.

3.1.3 Protein-protein Interaction Data

We used the same protein-protein interaction data as in the AC [27] method to make a fair comparison of our method with the AC method. Furthermore we applied the same procedure on the data set, to separate the data as training and test sets, as in AC method to carry out the experiments. Additionally we conducted the experiments for a second data set which includes a protein-protein interaction network with a larger number of interacting and noninteracting pairs of proteins.

3.1.3.1 First PPI Data

In AC method, Database of Interacting Proteins (DIP) was used to collect the PPI data of Saccharomyces cerevisiae organism for experiments. They generated the positive data set by getting rid of the proteins composed of less than 50 amino acids which resulted in 5943 proteins. Since the PPI network data in our experiments are constructed using the phylogenetic profiles of each protein, we made an extraction of each protein from the phylogenetic profiles in our hand. However the phylogenetic profiles data does not include the complete data set, that is to say not all of the proteins collected from DIP have the phylogenetic profile informa-

[2] ftp://ftp.genome.jp/pub/kegg/genes/taxonomy
[3] http://pir.georgetown.edu/pirwww/dbinfo/pir_psd.shtml

tion. Therefore the protein pairs including proteins without phylogenetic profile information are also eliminated from the data used in the experiments of the AC method. At last 5825 protein pairs left for the positive data set.

The interacting protein pairs can be obtained from several sources having various reliability measures, however the noninteracting pairs are not readily available. In AC method, the noninteracting pairs are obtained by 3 different ways and the performance of their methods by executing the experiments with entire of the negative sets is observed. The methods to generate the non interacting proteins data set are:

- Randomly pairing proteins from the positive data set. (The data generated by this method is named Prcp.)
- Pairing proteins occurring in different subcellular localization information. (It is assumed that the proteins that take place in different localizations in the cell do not interact [27]. The subcellular localization information was taken from the SwissProt[4] database. The data generated by localization information is named Psub.)
- Shuffling the protein sequences in the positive set. (When the protein sequences of two interacting proteins are shuffled it is be assumed that the two proteins do not interact with each other.)

At the beginning we included all the negative data sets to the core training data set and applied random data selection method on them. We observed that except the Psub data the prediction results of SVM with the training data including the other two negative data sets were rather inconsistent such that although the data size got linearly larger, the results were severely with ups and downs. On the other hand, the training data including the Psub negative data set gave reasonable results as we expected such that the prediction accuracies got linearly larger as the data grew in the same form. That is why we chose to use the Psub negative data by excluding the Prcp and the one generated by shuffling as the negative set.

In the AC method, the sizes of positive and negative data sets are kept the same. Therefore they produced 5943 protein pairs that do not interact. As in the positive set case, we searched for the phylogenetic profiles of the proteins in the negative set. Again there were

[4] http://www.expasy.org/sprot/

some proteins in the negative set that do not have the phylogenetic profile information in the phylogenetic profiles that we produced. After we eliminated the ones whose profiles were absent, 5871 protein pairs for the negative set are left.

In our experiments the sizes of positive and negative set are not exactly the same as they were in AC method but the difference is so ignorable that it would not affect the prediction results of SVM.

3.1.3.2 Second PPI Data

A second data set is generated to test the methods in a larger data set and observe how the results will differ in a larger interval of data set sizes. The positive pairs for second data set is constructed by making use of the interactions in the DIP database and the negative pairs are obtained from the MIPS database by randomly selecting the negative pairs at different subcellular localizations. This time the sizes of positive and negative data are not the same, instead the size of negative set is four times larger than the positive data. There are 17514 positive and 71231 negative pairs were in the original dataset. After the extraction of the phylogenetic profiles of proteins in each set and discarding the proteins whose profiles are absent we have 16987 positive and 67849 negative protein pairs in the second set of data.

3.1.4 Training and Test Data

We repeated the construction of the training and test set processes as done in the AC method. The final set for the first data set includes 11814 protein pairs and for the second data set 88745 protein pairs in total. The training set is for SVM learner is constructed from the three fifth of the whole data set. The remaining two fifth of the whole set comprises the test set. This operation is applied on the complete data set for five times and the separation of the data set is conducted randomly. Therefore a five-fold cross validation is used to investigate the training set [27] by applying the methods for each of the five samples. This process is pointed out at the top of the flow in the Figure 3.7

The aim of sampling the training set for five times is to prevent that the results produced from any method are particular to the characteristic of the training set it tested. By using 5 different training and test sets we observe the results of the methods for the data inputs exhibiting

different characteristics. Also the sampling of the data set is a way to observe whether the results of the methods exhibit the same characteristic for all samples. Therefore it would be proven that the methods produce deterministic and reasonable results. At last, the average of the five prediction results are calculated to represent the performance of the method.

The experiments aimed to compare the clustering methods are experimented on a data set whose training and testing subsets exhibit the same biological behaviour. By this way the changing in performance of each method would be observed when the data size is incremented linearly. For each sample in the first data set, the training data portion is sampled for 10, 20, ... 100% of the complete training set whereas the size of the test sets are kept the same. The sampling of the data, i.e. the data reduction, is conducted by three methods which are described in detail in Section 3.4. For the second data set, since the memory and hard disk were insufficient for sampling not all of the 10 samples for each experiment could be produced.

3.2 Learning and Making Prediction on Protein-Protein Interactions

A stated earlier, the protein-protein interaction prediction process is conducted by means of an already constructed interaction network where the coevolutionary knowledge of proteins are considered. SVM algorithm is selected for the aim of learning the known subset of the protein interaction network in order to make use of the coevolutionary data as a whole, i.e. avoiding to data loss. Because SVM is a method which provides a way to preserve the specialities of the data set. In the following subsections we present how we made use of SVM and its parameters in our experiments. RBF kernel is chosen for some reasons one of which is it maps the data into a higher dimensional space, so unlike linear kernel it can predict the class labels well when the relation between the class labels and attributes are nonlinear [4]. Linear kernel is a special case in RBF kernel since as stated by Keerthi and Lin [16] linear kernel with a penalty parameter C can achieve the same performance with RBF kernel with some C, γ. Also sigmoid kernel also can give reach the same results by some parameters. There are various kernel functions supplied, even the functions can be developed and fed by the user in SVM. In our experiments we chose to use RBF kernel since it mostly adapts the properties of our data. Problem of predicting protein-protein interactions fits cleanly into a binary classification framework where SVMs discover whether a given pair of proteins inter-

act or not. The essential question is how to represent the protein pairs [8]. The phylogenetic profiles constitutes the feature vector supplied to SVM and since the interaction network does not include any edge weight, the labels of the data are binary. The feature vector construction will be mentioned in Section 3.3.

3.2.1 Parameter Selection in SVM

When running SVM, the C and γ parameters are significant to choose since different values may effect the prediction accuracy values. That is why, it is essential to choose the C and γ parameters which yields the optimum resulting values. The grid operation of libsvm is run for the concatenated and ”exclusive or”ed profiles of the two data sets in order to reach the C and γ parameters that gives the most accurate predictions.

3.3 Integration of Pylogenetic Profiles for Feature Vector Construction

We applied two methods to integrate two phylogenetic profiles for the aim of constructing the feature vectors which are concatenating the profiles and applying exclusive or operation on the profiles. The integration of phylogenetic profiles, as in Figure 3.1, is achieved by first extracting the phylogenetic profiles of proteins individually which are represented by their ORF names, then applying concatenation or exclusive or operations on the profile pairs both for positive and negative interactions.

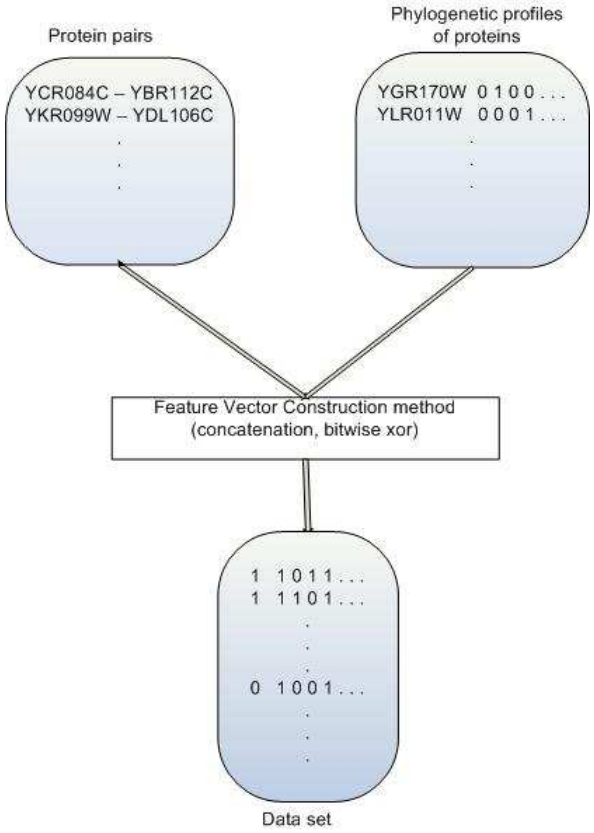

Figure 3.1: Construction of data sets by integrating phylogenetic profiles

3.3.1 Concatenation of Profiles

The phylogenetic profiles are concatenated by directly adding the two profiles one after the other. So in the concatenation of profiles method, the length of a feature vector becomes 2*number of organisms, i.e. 900 in our case. Therefore the set of organisms are repeated after the 450^{th} index. This method is aimed to keep the values in the two phylogenetic profiles without losing any information in the feature set. Though the running time of training with concatenated feature vectors is rather long, its prediction accuracy performance is superior to taking exclusive or (xor) of two profiles method which is described below.

3.3.1.1 SVM Parameters for Concatenated Profiles

A grid search tool which is provided by the libsvm-2.84 package is employed to obtain the optimum C and γ values which are used in the training phase of SVM. Below figures are the

contours generated by the grid.py for each of the training data generated from the first data set for the five experiments The center of the innermost frame gives the corresponding parameter values:

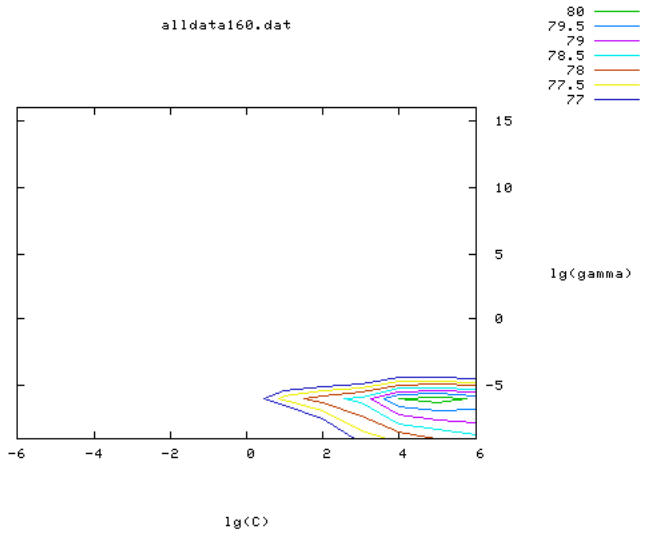

Figure 3.2: C and γ parameters for concatenated profiles of first experiment

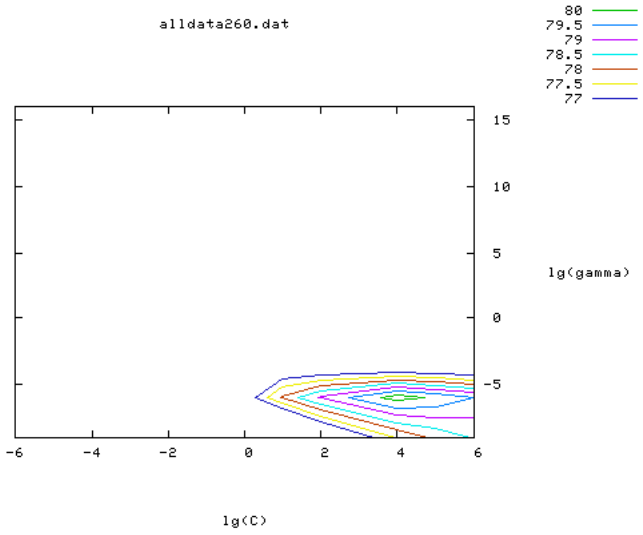

Figure 3.3: C and γ parameters for concatenated profiles of two experiment

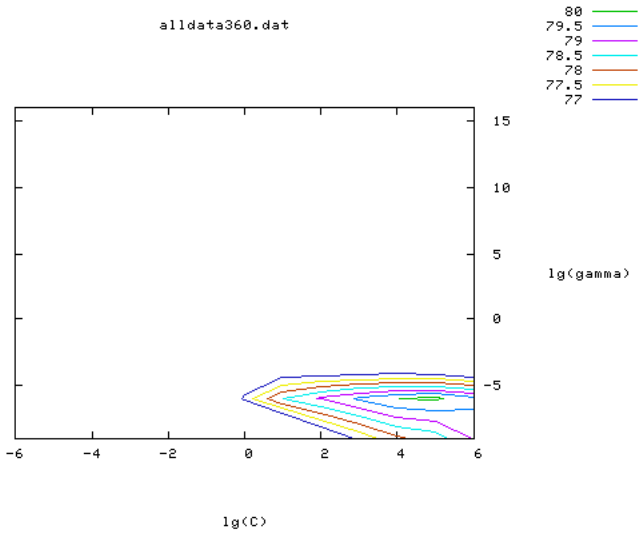

Figure 3.4: C and γ parameters for concatenated profiles of third experiment

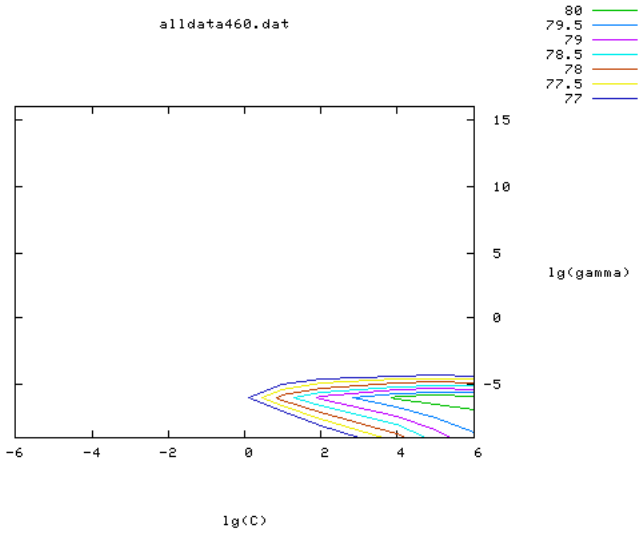

Figure 3.5: C and γ parameters for concatenated profiles of fourth experiment

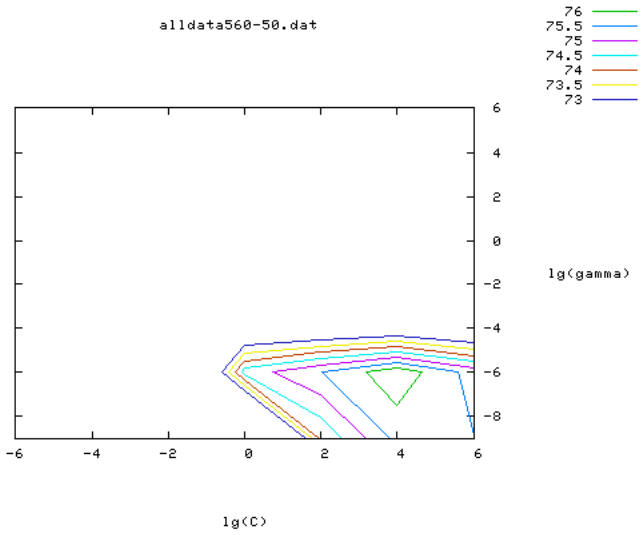

Figure 3.6: C and γ parameters for concatenated profiles of fifth experiment

The C and γ values can be obtained from center of the innermost frames in the contours, still grid search tool provides the exact values in a source file. For the first four experiments we obtained the same values which are $C = 64$ and $\gamma = 0.015625$ and for the last experiment the C parameter took the value 16 and the γ stayed the same.

3.3.2 Applying Bitwise Exclusive Or (Xor) Operation on Profiles

In bitwise xor operation method, the two profile values in the same index are xor'ed with each other. Bitwise xor operation which gives 0 for the same input values (both 1 or both 0) and 1 to different input values, is a way to yield the difference of two profiles to SVM. Hence the phylogenetic profile values possessing the same values will acquire the value 0 in the resulting profile and the different ones will result in 1 which means the profile elements having 1 and the ones having 0 as value will be treated the same in the resulting feature vector. The outcome vector after xor operation has the same length with the two profiles which is 450 in our case. The xor operation between two bits is represented as $a \oplus b = (a \cdot \overline{b}) + (\overline{a} \cdot b)$ where $\cdot$ and $+$ are *and* and *or* operations respectively and $\overline{a}$ and $\overline{b}$ are the inverses of a and b.

The training data constructed by the bitwise xor of the profiles could provide information about the resemblance of the two profiles, however since the weight of a feature vector member constructed from two 1's or two 0's will be the same for SVM, there might be some data loss in the resulting feature vector.

3.3.2.1 SVM Parameters for Exclusive Or'ed Profiles

The parameter values for the first training data set constructed by making xor operation between the two feature vectors of paired proteins obtained are represented in the following table.

The C and γ values for all the samples of the second data set are taken as 32 and 0.015625 respectively since it takes too much time for the cross validation process to end. These two values were also used as the optimal values in the experiments conducted in the AC method.

Table 3.1: C and γ values for xor'ed profiles

	C	γ
1^{st}	32	0.015625
2^{nd}	8	0.015625
3^{rd}	8	0.015625
4^{th}	64	0.015625
5^{th}	64	0.015625

3.4 Data Size Reduction

To reduce the training time of SVM while preserving the prediction accuracies for the interactions, two data clustering methods and a random selection model are experimented and compared. K-means clustering is one of the well-known and simplest clustering methods in literature for obtaining a certain number of clusters in a data set. On the other hand another data clustering method, named as Minimum Enclosing Ball Clustering is developed with some modifications for reducing the data by selecting the most significant subset for SVM training.

3.4.1 Random Data Selection

Random selection is applied as a baseline method to demonstrate the significance of the other clustering methods on the data in terms of accuracy. It is expected that the accuracy performance with the data set which is randomly selected, produces worse results compared to the other methods. We sample the training data sets of each of the five samples for the size of the 10,20..100% over the whole training set each of which are obtained randomly.

3.4.2 Data Selection by K-means clustering

There might be some elements in the data that are more representative than the ones in the rest of the data in that they can exhibit the characteristics of the whole data better than the others or some elements can have the approximate significance value in terms of their representativeness. In that case the elements have no benefit to be added into the training data, instead adding the data which characterize the data no better than the others, increases the training time. On the other hand some elements that mischaracterize the data set can exist. Discarding the noisy data if there are any, is a considerable factor for enhancing the accuracy.

K-means clustering technique is applied on the training data to gather more representative elements from the data set. The experiments are done by picking the data closest to the centroids of the clusters to generate the training set since data in the centroid is the one having minimum distance to the other elements in the cluster and having the most representative value among others. In k-means algorithm the distance between two points are calculated with Euclidean distance metric. The number of clusters (the number k in the algorithm) is selected as the number of data desired to be generated, i.e to generate the subsets with 10-100% of the whole set, k is picked as the data size for that run.

In the experiments for the first data set, the number of positive and negative data are desired to be kept equal to be consistent with the AC [27] method. That is why the k-means clustering method is applied on the positive and negative data separately, then the centroid points from the two sets are associated to construct the training data set.

3.4.3 Data Selection by Minimum Enclosing Ball Clustering

A novel approach for the reduction of data size is developed in MEB clustering algorithm. We then make a small modification in the algorithm and compare its result to the other techniques.

3.4.3.1 MEB Clustering Algorithm

The data size reduction algorithm proposed by [27] finds the smallest ball in a data set which includes all the points and uses the core sets idea [10, 11] to generate clusters in the data. Then using Sequential Minimal Optimization(SMO) idea that they proposed, they add the support vectors which are cluster centers also.

Below are some of the definitions used in the MEB clustering algorithm as stated in [6] :
$B(c, r)$ is the ball with center c and radius r.
The MEB of the set of points $S = x_i, ..., x_m$ is the smalest ball that contains all the data in S that is denoted as $MEB(S)$.
An approximation is applied since it is difficult to calculate $MEB(S)$ accurately. $(1 + \varepsilon)$-approximation of $MEB(S)$ is the ball denoted as $B(c, (1 + \varepsilon)r)$, $\varepsilon > 0$ with $r \geq r_{MEB(S)}$ where $S \subset B((1 + \varepsilon)c, r)$.
To approximate $MEB(S)$ with $(1 + \varepsilon)$ factor, k balls $B_1, B_2, ..., B_k$ are obtained where $S \subset$

$B_1 \cup B_2 \cup ... \cup B_k$.

They make a guess about the number of clusters using the support vector number of the set. They propose that the optimal number of clusters is $l = \frac{2}{3}sv$. The steps of the MEB algorithm are as follows:

- First of first, they pick the l ball centers $C = c_1, c_2, ..., c_l$ randomly and they use the same radius r for all of the balls by a method that they proposed in their paper.
- Calculate the Euclidean distance of each data point to the center of the ball they belong to. $\varphi(x_i) = \|x_i - c_k\|^2$ where $k = 1, 2, ..., l$ and select the point with maximum distance any cluster. If this point is not inside any cluster, then continue with fourth step.
- Increase the radius of balls by δ, $\varepsilon = \varepsilon + \frac{r}{\delta}$ until all the data is included in the balls
- Clustering is done with $B = (c_k, (1 + \varepsilon)r)$

After the first clustering step is ended the data set is separated into l partitions. Then a binary SVM classification follows to extract the support vectors in the data set. At this step the data reduction process starts. After the SVM classification, all the points are assigned a label. When the labels of data in a cluster differs, then only the center of the ball is included and the rest of the cluster is ignored. If all the elements in a cluster possess the same label, all the elements in that cluster are included. So the new data set is generated by $C^+ \cup C^- \cup \omega_m$ where C^+ and C^- are the centers of the clusters with all the elements having + and − labels respectively and ω_m represents the elements in the clusters with mixed labels. Next a declustering is applied by including the data points which are also cluster centers. This step raises the data size but enhances the accuracy of SVM. At last a second stage SVM is applied on the final training set.

3.4.3.2 Modified Minimum Enclosing Ball Clustering

Due to some uncertainties in the description of generation of balls in the MEB clustering algorithm, we made some changes in first stage of the algorithm. Instead of separating the data by partitioning by balls, we applied k-means clustering on the data. We used the same number of clusters as in the MEB algorithm, i.e. $\frac{2}{3}sv$ and we applied the rest of the algorithm as it is. So the steps that we performed are:

- Separate the data by k-means clustering with $k = \frac{2}{3}sv$.
- Apply SVM on the data set to assign labels to each element.
- Include the the whole data in the clusters with elements having same labels and only the elements in centroids of the clusters with mixed labels.
- Include the support vectors which are also cluster centroids.
- Apply the second stage SVM to the reduced data.

The Modified Ball Clustering method is applied on the whole training data set instead of clustering the positive and negative sets separately as in the k-means case because it is difficult to infer the number of data the MEB algorithm will return even if an initial number of clusters is given approximately at the beginning of the method.

3.5 Feature Selection

The data reduction is also applied on the feature vector size by picking the feature vector elements wisely to preserve the same accuracies with the experiments done by the complete feature vectors. We experimented three data reduction methods where a random feature selection is tested to monitor the differences of other methods in performance.

3.5.1 Random Feature Selection

As in the case with random data selection, random feature selection is applied on each feature vector to observe the effectiveness of the other feature selection methods in terms of accuracy. In feature selection using phylogenetic trees, approximately 250 feature vectors were left after the selection process. The same number of features with the random selection were picked among 450 organisms for a fair comparison of the methods.

3.5.2 Clustering Organisms by Phylogenetic Trees

The phylogenetic tree that we used to make a clustering among the organisms was composed of four levels. We cut the tree from a designated level and under each node in the cut level

there is a grouping of organisms which lay in the leaves. We picked the organisms from each group which have smallest distance to the other organisms in that group. The organisms that we selected are the most representative ones of the entire organisms set genetically. We experimented to cut the tree from each level and we obtained the most optimum results from the third level. Table 3.2 represents the number of organisms generated for each cut level.

Table 3.2: The number of organisms for each cut level

cut level	Num. of organisms
1	60
2	85
3*	250
4	438

In first and second levels most of the organisms are eliminated which causes some data loss, while in the fourth level only a very small portion of the organisms are eliminated which causes the feature vector not to be decreased in size sufficiently. After the most significant elements under third level are accumulated together to form the feature vector, the feature vector length becomes 250 which is an adequate decrease in size.

3.5.3 Fisher's Exact Test

We used Fisher's exact test to assign p-values to each of the feature vector elements. The p-values of a feature vector element gives a measure about how the element (i.e. the organism corresponding to that element) classifies the data. We set the p-values of the features in order and pick the first 250 to construct the feature vectors while making the length of the feature vectors same with the random selection and phylogenetic tree clustering methods. As an example below are the contingency tables and the p-values of two feature vector elements which are the 100^{th} and 308^{th} organisms respectively .

As seen in the tables, the p-value of 308^{th} element is fairly larger than the p-value of 100^{th} element, therefore, 100^{th} organism is excluded from the feature vectors.

250 organisms obtained after feature selection by Fisher's test can be viewed in Appendix A.1.2.

Table 3.3: Contingency tables for the 308^{th} and 100^{th} elements respectively

Label	Avail.	Unavail.
1	1130	4695
0	1835	4038

p-value=0.0902

Label	Avail.	Unavail.
1	39	5768
0	39	5834

p-value=6.1844E-50

3.6 Experimental Work

3.6.1 Data Size Reduction

The sequence of experiments for data size reduction techniques are combined together in the Figure 3.7. After the data set construction phase which was described in Figure 3.1, the whole data set is separated into 2 parts to compose the training and testing partitions. The training part is generated by randomly selecting the 60% of the whole set and rest of the data, 40% of the whole set constitutes the testing part. This operation is repeated for five times to obtain 5 different training-testing sets.

For each training-testing sample, we applied the same experimental procedures. The training data is sampled into sets with different sizes (10,20,30,...,100% for the first PPI and 10,50,100% for the second PPI data) using the two of the data reduction techniques which are random sampling and data selection by k-means clustering. For modified MEB clustering, the sampling is conducted for once since the data size after the reduction process can be predetermined. SVM models for each training sample in each training set are generated by training SVM. At last, the same test set is used to make predictions using these models to see the incremental changes in the prediction values for different sized training sets. The accuracy, true positive rates (TPR) and false positive rates (FPR) are calculated for each prediction result.

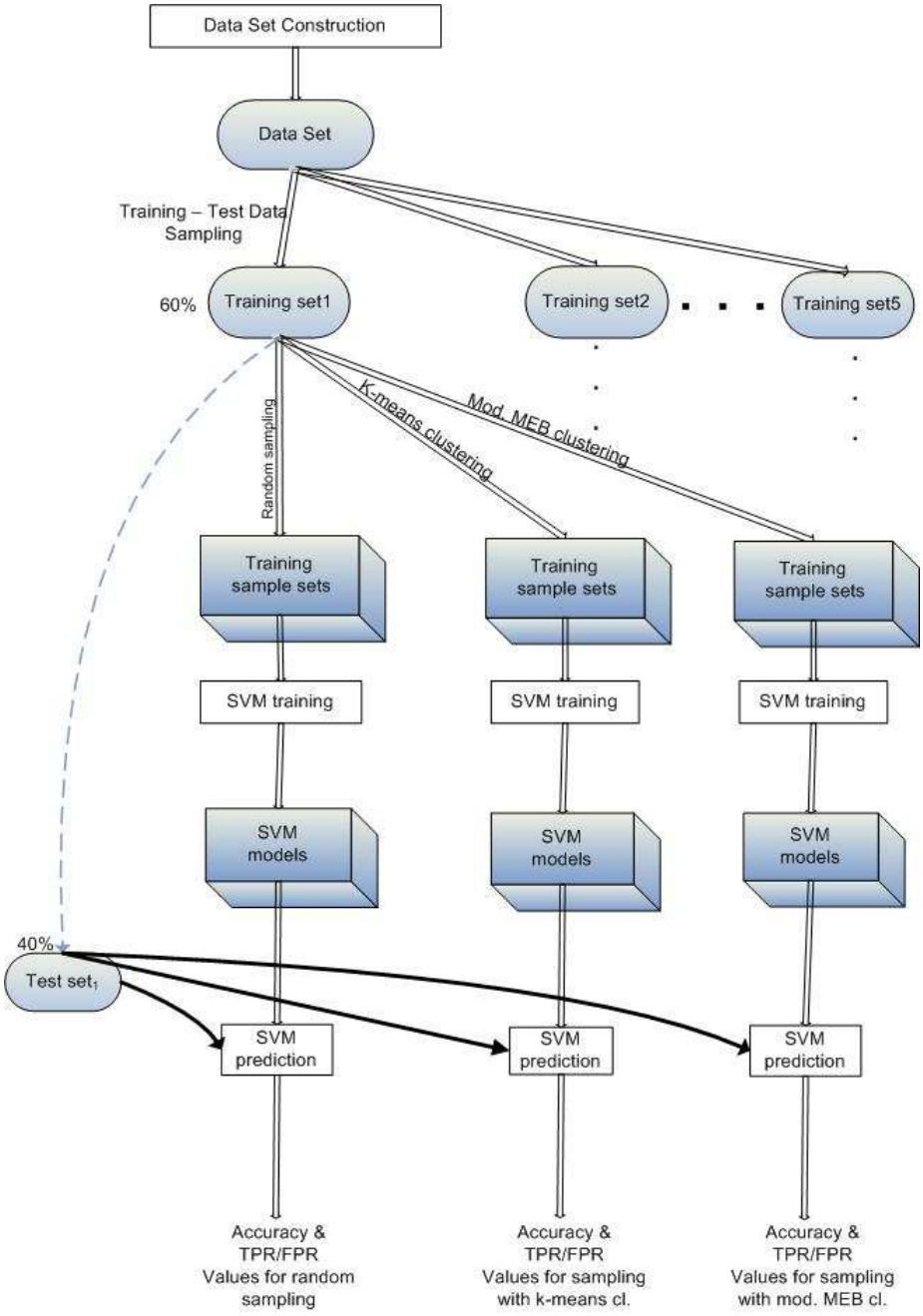

Figure 3.7: The flowchart to represent the sequence of methods applied in the experiments for data size reduction

3.6.2 Feature Selection

We applied the feature selection methods only on the training samples generated by the k-means clustering. The same experimental procedures are conducted on every training-testing sample as in the data sampling case. The flow diagram for feature selection is represented in

Figure 3.8 for n^{th} the training-testing sample where $n = 1, 2, .., 5$. The feature vector sizes in each data sample are reduced by performing three feature selection methods, i.e random selection, selection by phylogenetic tree and selection by Fischer's test, thus three different training sample sets are generated. The feature selection is also carried out on the test set since the training and testing sets must have the same feature size to make a proper prediction. Although the feature vector sizes are the same for each training sample after each feature selection method (250 for each which is described in Chapter 4), to select the same organisms for the training-testing samples, the test set is sampled for each method. After we obtain the models for each training data, the prediction is done for the corresponding SVM model. Once again the accuracy, TPR and FPR values are calculated for each prediction result.

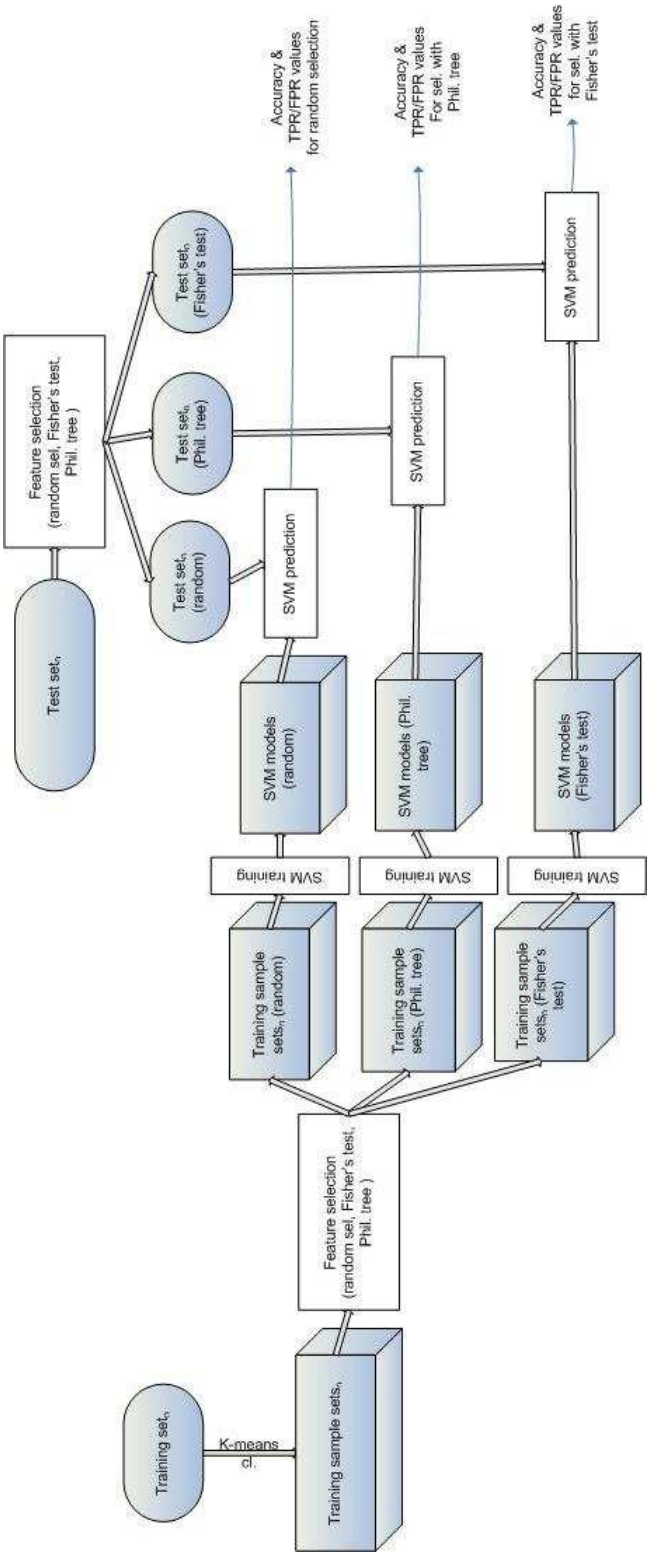

Figure 3.8: The flowchart to represent the sequence of methods applied in the experiments for feature selection

CHAPTER 4

RESULTS

In this section we present the two data sets used in the experiments then pointed out the experimental results of our methods together with our comments and explanations on the results.

4.1 Data Size Reduction Results

For the first data set the data samples for the 10,20,..100% of the whole training set are used for training to compare the results of random sampling and k-means clustering methods. To make the comparison between the performance of k-means and modified MEB clustering methods modified MEB clustering is run for once for the whole data set because the size of the training set comprised by the ball clustering method can not be previously determined. Then k-means is run once more for comparing the prediction accuracies of k-means and modified MEB clustering methods with same number of data. The k parameter in k-means is picked as the size of the data which is the output from the modified MEB clustering method. The prediction accuracies and the ROC curves indicate that the data size reduction with k-means clustering method exhibits higher performance than random selection and modified MEB clustering methods. The modified MEB clustering technique is applied only on the complete training data sets, i.e. the 100% of the training sets are sampled once for MEB clustering since size of the data generated by MEB clustering is not known before the experiment even if we give the number of clusters to generate in the algorithm. In the second stage of modified MEB clustering, we observed that very few data was added to the data set generated by the first stage. That is to say there are a few number support vectors in the training sets which are also cluster centers.

Since the training and testing data belong to the same data set actually, they represent the same characteristic of data. Therefore, while there are considerably large intervals among the data sizes, the prediction accuracies do not exhibit a quite much difference.

The points in the ROC curves are obtained by finding the TPR-FPR values of each data sample which are 10-100% of the training data. As the data size grows, the TPR also increases whereas the FPR decreases. In the graphs the data sizes are smallest for the rightmost points and largest for the leftmost values.

The samplings for feature selection methods are done over the samples obtained by k-means clustering by selecting 250 of the profiles.

The results of the second PPi data are given by tables instead of graphics. There are some empty cells in the tables representing the results for the second data set. The empty cells are because of the samples of the second data set which are not produced due to the memory or space insufficiencies. The second PPI data can not be sampled for all the percentage values used in the first PPI because of some lack of space. Therefore the random sampling is done for 10, 50% of the training data. On the other hand, 50% of the data for sampling with k-means clustering could not be obtained due to the lack of memory and we settled for sampling 10% of the training data in k-means clustering. Note that, because of this memory issue, the modified MEB clustering samples are taken over the 50% of the training data which we obtained randomly. Due to the insufficiency in memory for the experiments done by the second PPI data, the comparison between k-means and modified MEB clustering methods are done on the data samples constructed by randomly selecting 50% of the whole training data. Therefore the values for the comparison between the two clustering methods in the tables, belong to the experiments applied on the 50% of the training sets.

4.1.1 First PPI Data

4.1.1.1 Xor'ed Profiles

The average of accuracy results for the five samples of the first data set which are constructed by applying xor operation on the profiles are as in Figure 4.1

It is rather clear that the average of the accuracies for the five experiments of samples con-

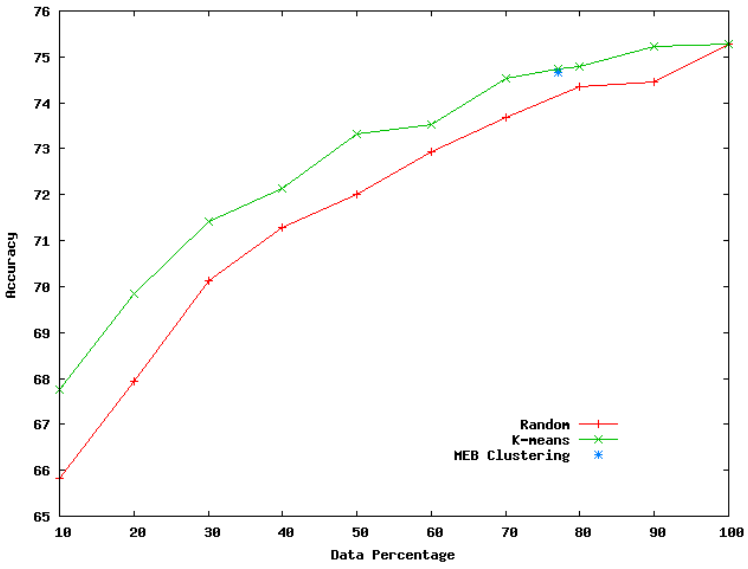

Figure 4.1: Accuracy results for the first PPI data constructed by xor operation

structed by k-means clustering have higher values than the samples randomly generated. The size of the data generated from modified MEB clustering is 77% of the whole training set size. Therefore, in order to make a fair comparison a k-means clustering is also applied to make a sample of 77% of the training sets. It is seen in Figure 4.1 that the accuracies of k-means and modified MEB clustering are very close to each other. However, in k-means clustering there are some empty clusters generated which causes the data size to be smaller than the number of clusters, i.e. the k value. Therefore the accuracy values calculated for each data size are actually belong to data samples with smaller data sizes. For instance 77% of the training data includes 5424 elements, but when k-means is run for 5425 clusters, the resulting data set includes 4984 elements which corresponds to 71% of the training set.

It is difficult to compare k-means clustering with modified MEB clustering with exactly the same sizes of data. Assuming that k-means gives at least the same accuracy with modified MEB clustering, we can conclude that, when run with exactly the same number of elements, k-means can give a higher accuracy than modified MEB clustering method.

Below is the figure for the ROC curves of the data samples generated by applying xor operation on the profiles for the three sampling method. According to both TPR and FPR values, the training data sampled by k-means clustering exhibits better performance, i.e higher TPR and lower FPR. In comparison between the k-means and modified MEB clustering for sam-

pling 77% of the whole training data, k-means results in higher TPR values whereas MEB clustering has lower FPR values. So it is not definitely observed which clustering method has better performance in TPR-FPR comparison.

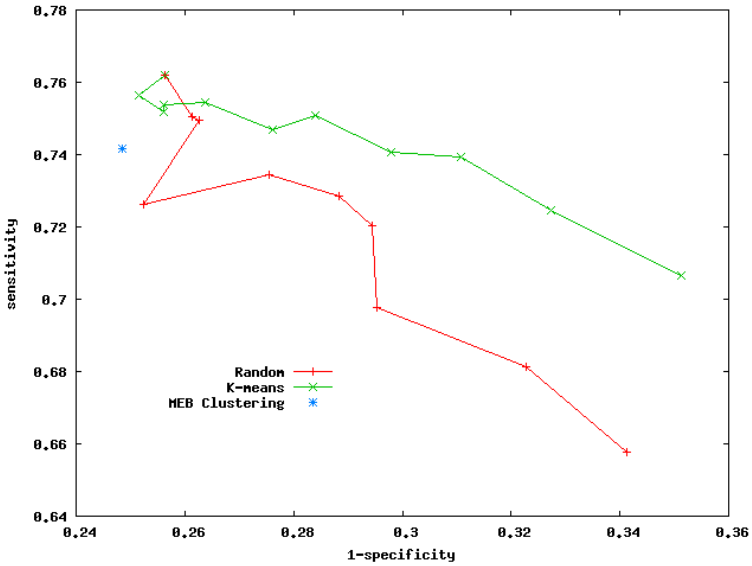

Figure 4.2: ROC curves for the first PPI data constructed by xor'ing

4.1.1.2 Concatenated Profiles

The average of accuracy results for the five samples of the first data set which are constructed by concatenating the profiles are as in Figure 4.3.

The accuracy values of k-means clustering is superior to random sampling for each data sample as in the xor'ed case. Furthermore the performance of k-means exceeds modified MEB clustering for the data size which is 73% of the training set. Again k-means is run for the k value corresponding to 73% of the data which is 5123 clusters, but due to the empty clusters generated, 4920 elements are left from the k-means run. In spite of smaller training data size, k-means algorithm achieved a higher performance than modified MEB in accuracy.

The ROC curves in Figure 4.4 clearly represents the achievement of k-means clustering method over random data selection since as the data size gets larger, the points in the graph gets higher TPR and lower FPR values which brings the points belonging to k-means clustering closer to left and upper corner of the graph when compared to random data selection. Likewise, k-means reaches a better TPR-FPR value than modified MEB clusering. So, accord-

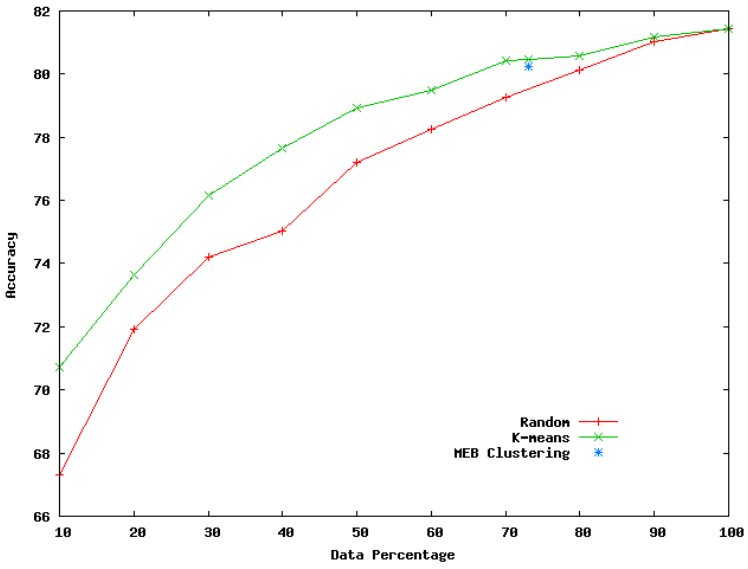

Figure 4.3: Accuracy results for the first PPI data constructed by concatenation

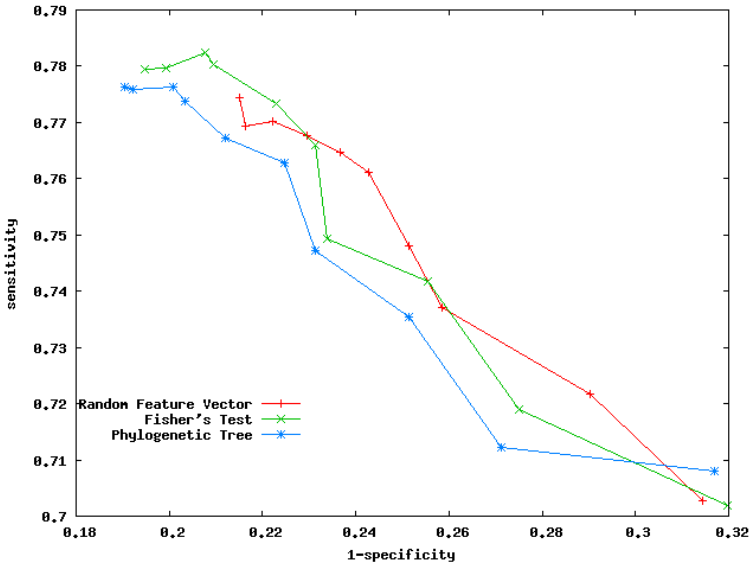

Figure 4.4: ROC curves for the first PPI data constructed by concatenating

ing to the experiments conducted by using the training data generated by the concatenation of profiles, the best performance is achieved with training data clustered by k-means clustering both in accuracy and in ROC curves.

4.1.2 Second PPI Data

4.1.2.1 Xor'ed Profiles

The prediction accuracies for the training data constructed by the xor'ed profiles of second data set are given in the below table. The k-means accuracy results for 10% of data have higher values than the random case as expected. When modified MEB clustering is run on the 50% of the training set, the resulting training data sample includes 67% of the half of the training data. K-means also reaches a higher accuracy value for the 67% data sample.

Table 4.1: Accuracy results for the second PPI data constructed by xor operation

%	Random	K-means	Mod. MEB Cl.
10	78.4506	**79.4018**	
50	80.5298		
67		**80.0104**	79.9735
100	81.4616	81.4616	81.4616

K-means clustering algorithm shows lower TPR performance than random selection whereas it has lower FPR values on 10% of the training set. While k-means results in a lower FPR value than modified MEB clustering, it exhibits a worse TPR value with 67% of the half of the data. Thus a definite comment can not be made according to the TPR-FPR values to make a comparison between the two clustering techniques and between random selection and k-means selection.

Table 4.2: TPR-FPR Results for the second PPI data constructed by xor operation

	Random		K-means		Mod. MEB Cl.	
%	TPR	FPR	TPR	FPR	TPR	FPR
10	**0.2227**	0.0748	0.2210	**0.0625**		
50	0.2726	0.0613				
67			0.2897	**0.0710**	**0.3046**	0.0763
100	0.2937	0.0549	0.2937	0.0549	0.2937	0.0549

4.1.2.2 Concatenated Profiles

The size of the data sample generated by the modified MEB clustering of the randomly selected half of the training set is the 67% of the half of the data as in the xor case above. As represented in Table 4.3 the samples of second data generated by concatenation gives the

higher prediction accuracy values in k-means clustering when compared with both random selection method for 10% sized data and modified MEB clustering for 67% of the data.

Table 4.3: Accuracy results for the second PPI data constructed by concatenation

%	Random	K-means	Mod. MEB Cl.
10	78.8401	**79.9045**	
50	81.7757		
67		**80.8487**	80.5452
100	83.1861	83.1861	83.1861

K-means outperforms random data selection in both TPR and FPR values calculated by using the 10% of the training data. However, in comparison of the clustering algorithm performances, k-means has a better FPR performance and modified MEB clustering shows a better TPR value which means it can not be figured out which clustering method outperforms the other by looking at the TPR-FPR values for the 67% of the half of the training set.

Table 4.4: TPR-FPR results for the second PPI data constructed by concatenation

	Random		K-means		Mod. MEB Cl.	
%	TPR	FPR	TPR	FPR	TPR	FPR
10	0.2654	0.0806	**0.2775**	**0.0703**		
50	0.3645	0.0687				
67			0.3271	**0.0710**	**0.3729**	0.086
100	0.3712	0.1023	0.3712	0.1023	0.3712	0.1023

4.2 Feature Selection Results

The prediction accuracies of the feature selection methods could not be superior to the data reduction methods, furthermore the accuracy values in data reduction techniques can not be reached in feature selection but still we made the comparison among the feature selection methods. As expected random feature selection represented the worst performance among the three feature selection techniques. The phylogenetic tree method gave higher accuracy results in some of the experiments compared to the Fisher's exact test method. The reason of this difference can be interpreted that phylogenetic tree method considers the coevolutionary characteristic in the data while Fisher's exact test selects the feature elements according to only statistical measures. However, those overtaking values of phylogenetic tree method can not be protected in some of the experiments. Hence an interpretation on the comparison of the feature selection methods can not be definitely made.

4.2.1 First PPI Data

4.2.1.1 Xor'ed Profiles

The Figure 4.5 represents that Fisher's test method starts with the worst accuracy for the smallest data, but as the data size grows, its accuracy reaches to higher values. Still, the prediction accuracies of Fisher's test and phylogenetic tree methods are close to each other for xor'ed data sets. On the other hand, the training data sets whose features are randomly selected, gives the worst accuracy results when compared to the other two methods.

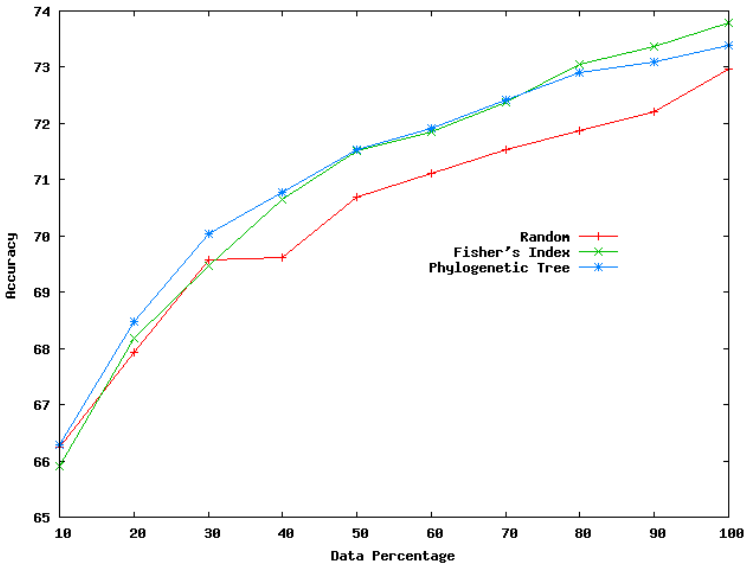

Figure 4.5: Accuracy results for the first PPI data constructed by xor'ing and with feature selection

For the ROC curves, it can be observed that Fisher's test and phylogenetic tree methods exhibits higher TPR and FPR performances especially for large data sizes compared to random sampling. But in general while the TPR values of Fisher's test and phylogenetic tree seems to be higher than those of random sampling the case is reversed for FPR values. Specifically for smaller training data sizes, FPR values of phylogenetic tree method represents higher values than those of others. The training data whose features are selected randomly results in the lowest FPR values except for the experiments done by the 100% of the training data.

According to the TPR-FPR values in the Figure 4.6 any specific interpretation to decide the best method can not be made. For TPR values of Fisher's test and phylogenetic tree methods

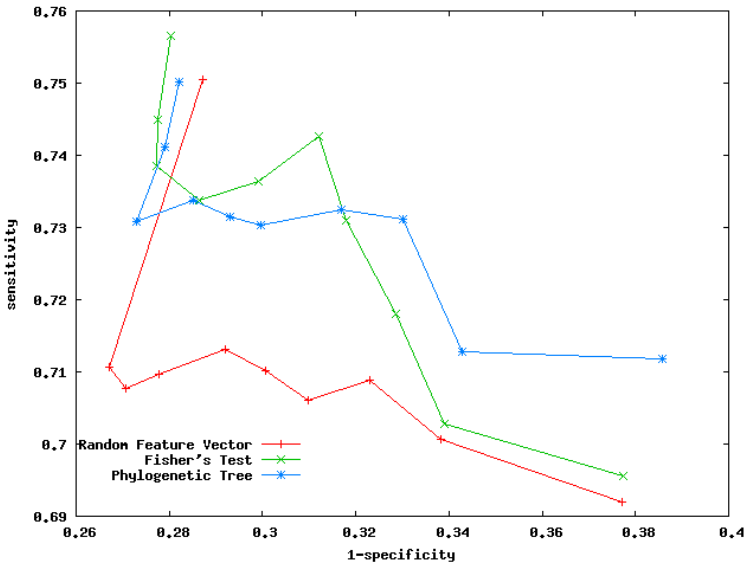

Figure 4.6: ROC curves for the first PPI data constructed by xor'ing and with feature selection

are close to each other for the average case while random feature selection have the lowest FPR values.

4.2.1.2 Concatenated Profiles

According to Figure 4.7 the prediction accuracies of Fisher's test and phylogenetic tree methods are very close to each other for all of the data sizes. It is clearly observed that random feature selection represents the lowest accuracy performances for all of the data sizes except for the 10% case. Therefore, phylogenetic tree and Fisher's test methods exhibit very close accuracy performances so that none is superior to the other.

In Figure 4.8, the Fisher's test represents the highest TPR values for the last 5 training data samples (i.e. 60, 70,..., 100%) whereas the TPR values of random feature selection reach the highest for the first 5 experiments compared to those of the other methods. The training data whose features are selected by the phylogenetic tree method has the lowest TPR values whereas, it has the highest performance in FPR values. Again the ROC curves of the three feature selection methods can not provide making a definite interpretation but we can only make comments about the performances of the methods over their TPR and FPR values individually.

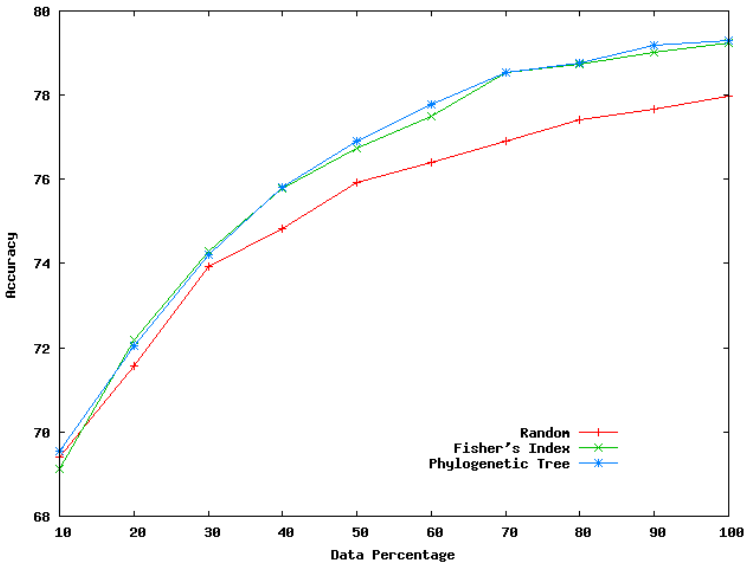

Figure 4.7: Accuracy results for the first PPI data constructed by concatenating and with feature selection

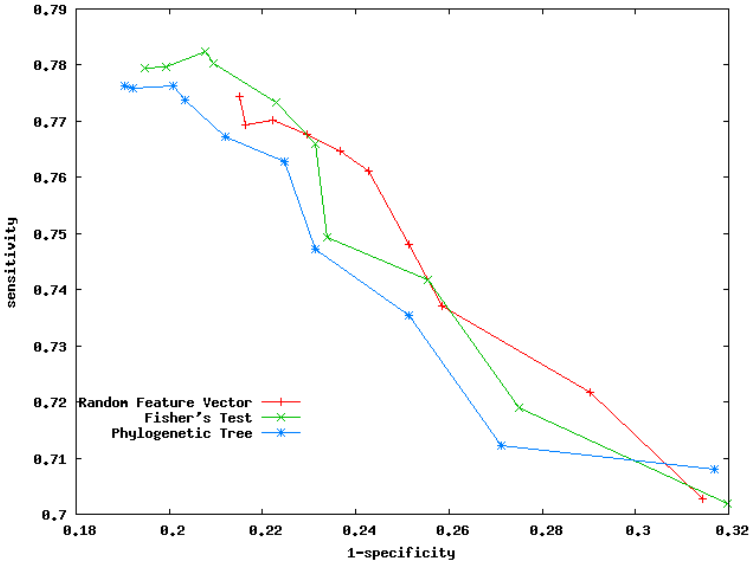

Figure 4.8: ROC curves for the first PPI data constructed by concatenating and with feature selection

4.2.2 Second PPI Data

4.2.2.1 Xor'ed Profiles

The prediction accuracies in Table 4.5 represent that randomly selected feature vectors of 10% of the the training data outperforms the other methods in accuracy. When trained by the whole

data set, the features selected by phylogenetic tree method gives the highest accuracy value among other methods and also Fisher's test method outperforms random feature selection for 100% of the training data.

Table 4.5: Accuracy results for the second PPI data constructed by xor operation and with feature selection

%	Random	Fisher's Test	Phyl. Tree
10	**79.1743**	78.9362	79.1142
100	81.1144	81.2606	**81.2701**

Random selection method is superior to other feature selection techniques for the FPR values obtained by training both of the data samples. Training by 10% of the data set whose features are selected by Fisher's test results in higher TPR value than the others whereas phylogenetic tree method applied on the whole training set produces higher TPR. We can not make an inference from the values in Table 4.6 to decide on the best method since the results do not provide a consistency with each other.

Table 4.6: TPR-FPR values for the second PPI data constructed by xor operation and with feature selection

	Random		Fisher's Test		Phyl. Tree	
%	TPR	FPR	TPR	FPR	TPR	FPR
10	0.1615	**0.0504**	**0.1752**	0.0568	0.1729	0.0540
100	0.2103	**0.0384**	0.2258	0.0404	**0.2275**	0.0407

4.2.2.2 Concatenated Profiles

The accuracy values in Table 4.7 obtained from the training sets which are formed by concatenation of profiles of proteins in second PPI data represents that phylogenetic tree method outperforms the other methods for both of the data sizes.

Table 4.7: Accuracy results for the second PPI data constructed by concatenation and with feature selection

%	Random	Fisher's Test	Phyl. Tree
10	79.4148	79.4383	**79.5567**
100	82.3619	82.6403	**82.7947**

The TPR values obtained from training by the data set whose features are selected by phylogenetic tree method exhibits the highest values among TPR values of other methods. However,

for the FPR values, as in the xor'ed profiles of second data set, the random feature selection method provides the best results. On the hand, the comparison of FPR values of phylogenetic tree and Fisher's test methods points out that phylogenetic tree has lower FPR for 10% of sample and Fisher's test has lower FPR values for the whole training data set. In spite that the accuracy and TPR values of phylogenetic tree outperforms Fisher's test, the case is reversed for FPR values.

Table 4.8: TPR-FPR values for the second PPI data constructed by concatenation and with feature selection

	Random		Fisher's Test		Phyl. Tree	
%	TPR	FPR	TPR	FPR	TPR	FPR
10	0.2161	**0.0611**	0.1839	0.0643	**0.2312**	0.0631
100	0.3148	**0.0490**	0.3300	0.0493	**0.3380**	0.0494

4.3 Prediction by Bayesian Learning

Bayesian Learning method is applied on the two PPI data to make a comparison with the predictions of SVM learning. In Bayesian learning the feature vectors are converted into a scalar value by applying a similarity measure on the profiles. We expect that this conversion causes some data loss in the training data since in SVM training we protect the co-evolutionary knowledge by keeping the profiles values. Therefore the prediction values obtained by Bayesian learning are expected to be lower than those obtained from SVM learning.

The similarity of two phylogenetic profiles are calculated by 3 methods; match count, mutual information, and hypergeometric distribution. First of first the similarity measures for the protein pairs in both the training and test sets are calculated. The protein pairs in training set are sorted according to these similarity scores. We calculate the positive and negative likelihoods for the predetermined intervals in the sorted list, i.e. the likelihood of interacting for each interval is calculated. The interval which results in the highest accuracy value is decided to be the decision threshold. Then the decision threshold is performed on the test set to obtain the accuracy values.

Below are the prediction accuracy results of Bayesian learning for the two PPI data.

4.3.1 First PPI Data

The prediction accuracies generated by the Bayesian learning results in lower values compared to prediction accuracies of SVM learning in the first PPI data considering both data and feature selection methods:

- match count: 0.58
- mutual information: 0.62
- hypergeometric distribution: 0.57

4.3.2 Second PPI Data

The second PPI data denotes approximate accuracy values with SVM learning except the experiment conducted by the similarities calculated by hypergeometric distribution measure. Furthermore, the accuracy values are superior to the values generated by training data obtained from the feature selection in SVM learning. Still it can be concluded that SVM learning using all the organisms to construct the feature vectors, i.e. without feature selection, reaches higher prediction values compared to those obtained from Bayesian learning.

- match count: 0.80
- mutual information: 0.80
- hypergeometric distribution: 0.53

4.4 Training Times for Data Sets Sampled from First PPI Data

Figure 4.4 includes the training times for the different sized training data which are sampled from some of the sampling methods described. The training times for the data samples generated by concatenation of profiles, xor of the profiles and feature selection by Fisher's test on concatenated profiles and feature selection by Fisher's test on xor'ed profiles are recorded. There is no specific reason for choosing the data samples with Fisher's test method since the data sizes generated by feature selection method are the same. The aim here is to measure

the training times for various sizes of training data. The number of training data, that is to say the number of interactions do not change for each sample of the 4 methods. Each sample constitutes the 10,20, ..., 100% of the training data set. The data size changes with the feature vector size which also affects the training time of SVM. The training data generated by the concatenated profiles is composed of feature vectors with size 900, whereas the feature vectors of the training data generated by xor'ed profiles are composed of 450 features. On the other hand, the training data with concatenated profiles which are seleced by Fisher's test include 500 features and the training data with xor'ed profiles and sampled with Fisher's test are composed of 250 features. As the interval between the data sizes grow, the training time of SVM also increases with larger intervals. Although training data with concatenated profiles exhibits the worst performance in training time, it has the crowning prediction accuracy values.

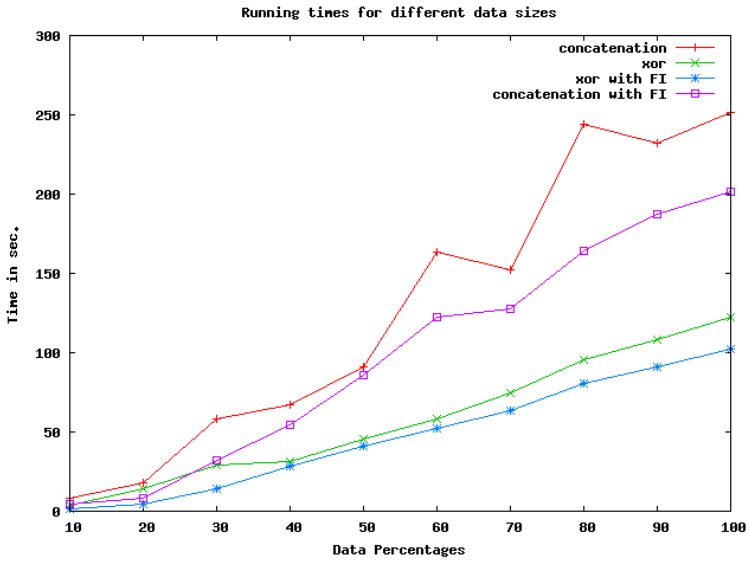

Figure 4.9: Running times for different sized data sampled from first PPI data

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Prediction of protein-protein interaction problems involving supervised learning may require the training time to be long. Since SVM solves the learning problem in quadratic time, as the data size grows it needs vast amount of time for training. In this thesis work, we proposed to use data reduction techniques to select the most representative data from the training set and compared prediction accuracy performances of these techniques. The phylogenetic profiles of the proteins, which incorporated the coevolution knowledge in the learning, constituted the feature vectors. For the construction of protein-protein interactions various data sources which are publicly available are exploited. The experiments are conducted for two data sets one of which is the same data set as generated in the AC method, that is another method using SVMs to predict protein interactions, and the other data set that we constructed having a larger size . We separated the data as the training and testing samples in same way described in the AC method to conduct the experiments. For all training data sizes, the prediction accuracy values generated by the training data with concatenated profiles achieves higher values than training data with xor'ed profiles.

For data reduction we applied k-means clustering and a modified version of MEB clustering techniques together with random data selection. In the experiments it was clearly seen that k-means clustering resulted in higher performance values in accuracy than the other two data reductions. But the TPR-FPR values did not give definite results to prove the dominance of one method over the other.

We applied the data reduction in the second dimension of the data by reducing the feature

vector size. We made a grouping among 450 organisms according to the third cut level of the phylogenetic tree that we used. In each group the organisms having the smallest distance to other organisms are selected and 250 organisms constituted the new feature vectors in the end. To make the comparison of the feature selection techniques with the same sizes, we applied the other methods on the training data, Fisher's test and random selection, by selecting 250 organisms. We observed that when feature selection applied on the training data, the same prediction accuracy values could not be kept as in the original training data which means even though the most significant features are tried to be picked, data loss could not be avoided. But when the feature selection methods are compared among each other, random feature selection displayed the worst accuracy values in most of the experiments but this is not valid for the TPR-FPR values, since random selection could outperformed the other methods in some of the experiments. A clear distinction between phylogenetic tree and Fisher's test methods, to decide on which one is superior, could not be made according to the the accuracy and TPR-FPR values since the experimental results are not consistent with each other most of the time.

5.2 Future Work

In the k-means algorithm, the resulting clusters change for each run since the results depend on the initial distribution of the centroids. It minimizes intra-cluster variance but does not ensure a global minimum of variance. In order to get better results, the initial distributions can be arranged systematically instead of randomly selection. Another alternative is that the algorithm can be run for several times until satisfying results are obtained.

Although k-means obtained relatively good results, to overcome the weaknesses of k-means other alternatives of k-means could also be tried such as k-medoids, fuzzy c-mean and k-mode. Apart from these other clustering methods could also be tried on the same data sets considering that other methods might discover the characteristics of the data sets better. These methods could be Self Organized Maps (SOM), hierarchical clustering, Gaussian Mixture or Learning Vector Quantization (LVQ).

In construction of the feature vectors 1 and 0 were used to dictate the availability of homology between two organisms. Instead of using binary numbers, floating point numbers could be

preferred to make use of homology by value instead of its availability. In this case the feature vectors would include multiple values which may cause the learner to run in longer time but may give more accurate results.

The prediction accuracies can be increased by including some additional data, i.e GO annotations of proteins, to the training data set. To achieve this some new feature vector construction techniques can be applied to manage to include multiple data.

Finally utilization of a protein-protein interaction network where the edges include weights can provide more detailed knowledge and enhance the prediction results.

REFERENCES

[1] Jothi, R., Przytycka, T.M., Aravind, L., (2007) Discovering functional linkages and uncharacterized cellular pathways using phylogenetic profile comparisons: a comprehensive assessment, Bioinformatics 2007, 8:173.

[2] Shen, J.W., Zhang, J., Luo, X.M., Yu, K.Q., Chen, K.X., Li, Y.X. and Jiang, H.L. (2007) Predicting protein-protein interactions based only on sequences information. Proc Natl Acad. Sci. USA, 104, 4337-4341

[3] Coward, E. (1999) Shufflet: shuffling sequences while conserving the k-let counts. Bioinformatics, 15, 1058-1059

[4] Hsu, C.W., Chang, C.C., Lin, C.J. (2008) A practical guide to support vector classification, Department of Computer Science, National Taiwan University

[5] Cortes, C., Vapnik, V., (1995), Support-Vector Networks, Machine Leaming, 20, 273-297.

[6] Cervantes, J., Li, X., Yu, W., Li, K. (2008) Support vector machine classification for large data sets via minimum enclosing ball clustering, Neurocomputing, 611-619

[7] Chen, J., and Yuan, B.,(2006) Detecting functional modules in the yeast protein-protein interaction network, Bioinformatics, 2283-2290.

[8] Vert, J,-P., (2001) Introduction to Support Vector Machines and applications in computational biology, DRAFT.

[9] Davis,J.,Goadrich, M., The Relationship Between Precision-Recall and ROC Curves, Department of Computer Sciences and Department of Biostatistics and Medical Informatics, University of Wisconsin

[10] P. Kumar, J.S.B. Mitchell, A. Yildirim, (2003)Approximate minimum enclosing balls in high dimensions using core-sets, ACM J. Exp. Algorithmics 8.

[11] Badoui, M., Har-Peled, S., Indyk, P., (2002) Approximate clustering via core-sets, Proceedings of the 34th Symposioum on Theory of Computing.

[12] Saitou, N. and Nei, M., (1987) The neighbor-joining method: A new method for reconstructing phylogenetic trees. J. Mol. Evol 4:406A425.

[13] Ref: Bowers PM, Pellegrini M, Thompson MJ, Fierro J, Yeates TO, Eisenberg D., Prolinks: a database of protein functional linkages derived from coevolution., Genome Biol. 2004;5(5):R35.

[14] Yohan Kim and Shankar Subramaniam, Locally Defined Protein Phylogenetic Profiles Reveal Previously Missed Protein Interactions and Functional Relationships, PROTEINS: Structure, Function, and Bioinformatics 62:1115n1124, 2006

[15] Vert, J-P. (2002) A tree kernel to analyse phylogenetic profiles, Bioinformatics Vol. 18 no. S276-S284

[16] Keerthi, S. S. and C.-J. Lin, (2003) Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation 15 (7), 1667Ğ1689.

[17] Sato, T., Yamanishi, Y., Kanehisa, M., and Toh, H. (2004) Prediction of protein-protein interactions based on real-valued phylogenetic profiles using partial correlation coefficient. GIW 2004 Poster Abstracts, P122

[18] D. Juan, F. Pazos, and A. Valencia, (2008) "High-confidence prediction of global interactomes based on genome-wide coevolutionary networks, PNAS, vol. 105, no. 3, pp. 934-939.

[19] Sun J, Xu J, Liu Z, Liu Q, Zhao A, Shi T, Li Y., (2005) Refined phylogenetic profiles method for predicting protein-protein interactions.,Bioinformatics. 15;21(16):3409-15.

[20] Sun J, Li Y, Zhao Z., "Phylogenetic profiles for the prediction of protein-protein interactions: how to select reference organisms?" Biochem Biophys Res Commun. 2007 Feb 23;353(4):985-91)

[21] Zheng, Y., Roberts R. J., Kasif, S., (2002) Genomic functional annotation using coevolution profiles of gene clusters, Genome Biology, 3(11):research0060.1-0060.9.

[22] Gonzales, O., Zimmer, R., (2008) Assigning functional linkages to proteins using phylogenetic profiles and continuous phenotypes, Bioinformatics. Vol. 24 no. 10 2008, pages 1257Ğ1263.

[23] Wu, J., Kasif, S., DeLisi, C., (2003) IdentiŞcation of functional links between genes using phylogenetic proŞles, Bioinformatics Vol. 19 no. 12, pages 1524-1530.

[24] [assign] Pellegrini, M., Marcotte, E.M., Thompson, M. J., Eisenberg, D., Yeastes, T. D., (1999) Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles, Proc. Natl . Acad. Sci. 4285-4288.

[25] C. Campbell, (2007), Studies in Fuzziness and Soft Computing, Chapter 7 (An Introduction to Kernel Methods), p. 162.

[26] Bock, J. R., Gough, D. A., (2001) Predicting proteinĞprotein interactions from primary structure, Bioinformatics Vol. 17 no. 5, 455Ğ460.

[27] 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 1-6

[28] Martin, S., Roe, D., Faulon, J.-L., (2005) Predicting proteinĞprotein interactions using signature products, Bioinformatics Vol. 21 no. 2, 218Ğ226.

[29] Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997), Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research, 25:3389-3402.

[30] Chang, C. C., Lin, C. J. (2001). LIBSVM: A library for support vector machines.

[31] Thompson, J. D., Higgins, G. D., Gibson, T. J., (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, Nucleic Acids Research, Vol. 22, No. 22 4673-4680.

[32] Notredame C., Higgins4, D. G., Heringa, J., (2000) T-coffee: a novel method for fast and accurate multiple sequence alignment, Journal of Molecular Biology Volume 302, Issue 1, 205-217

[33] Edgar, R.C., (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity, Bioinformatics, 5:113.

[34] Mering1,C., Huynen, M., Jaeggi D., Schmidt, S., Bork1, P., Snel, B., (2003), STRING: a database of predicted functional associations between proteins, Nucleic Acids Research, Vol. 31, No. 1

[35] Kanehisa, M., Goto, S., (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes, Nucleic Acids Research, Vol. 28, No:1

[36] Pagel,P., Kovac, S.,Oesterheld, M., Brauner, B., Dunger-Kaltenbach, I., Frishman, G., Montrone, C.,Mark, P., Stümpflen, V., Mewes, H., Ruepp A., Frishman, D.(2004) The MIPS mammalian protein-protein interaction database, Bioinformatics, 832-834.

[37] Xenarios, I., Salwinski, L., Duan, X. J., Higney, P., Kim, S.-M., Eisenberg, D., (2002) DIP, The Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions, Nucleic Acids Research, Vol. 30, No. 1, 303-305.

APPENDIX A

ORGANISMS

A.1 Organisms Used for Constructing the Phylogenetic Profiles

Below are the shortened names for the 450 organisms which are used to generate the phylogenetic profiles of the proteins of yeast organism.

bqu cfa eba sme rxy dpyo tan sma dcbr nme reh cyb cya nma jan tac dme cel ttj tth tte lpp cef lpn dbmo lpl vpa pho mxa xla bps ilo lpf hdu bpn bpm rde pha bpe bpa gox cdi aha rco osa hwa pgi ccr ddha ago hch mca erw dkwa eru cch wbr pfu wbm cca pfo cvi pfl erg mbo afu cbu rbe dtni pfa rba afm ayw sit sil lmo pen bms rty lmf lme yps tpv lma gme ypn mtu ypm ypk hal bmf bme mtp mac bmb ype bma cal aeh hac ypa ctr mth shm tpa cac cab mtc wsu she blo cte sha llc ctc dyli lla cta rsp nha msu bli rso bld hso pcu ade sgl pcr ngo ooe hsa rru csa xft pca gka sfx sfv dge sfu dncr sfr aci ace crp syw ljo sfl xfa uur syn dfru ava syg nfa syf sye syd abo syc bja ser neu sep pub net pau pat neq par lin tma aba lil dsba det dcnb sec pai lic ptr rpr pae pto mpu pac fal pab hpy sdy deh ftu mpn atu cpv cpt rpe rpd cps ftl aae sdn cpr rpc rpb rpa mpe hpj fth cpn ftf mpa cpj ath sde swo eli hpa cpf pst cpe atc psp cpa wol xcv ddi nwi tko bhe dde bha sco psb xcc xcb lga rno dvu gga bga art vfi sbo sus bfs hne bfr fra cne mmy bfl mmu mmr mmp sav mmo sau sat sas sar cmu rme sao san sty nar sam xtr oih sal sak lxx sai stt mmc xac mma sag sto sac dar ppu hma sab stm stl cme saa ppr sth ste stc mlo ppe dcin rle bxe mle dame poy aph ssp sso ehi ape ssn dspd ldb dsy ddpo pol ssc lwe vvy nse vvu aor dkla vch bcz sru mka dpkn bcn dmgr tfu bcl lca bci bch fnu bce bcc gvi dcgr bca lbu lbr cjr dre lbl bbu mja zmo lbj cjk ani dra bbr pmu pmt cje ana gbe pmn ter pmm efa bur pmi tel hit bba spz hin pma spy buc spt sps bat spr bas bar dsmi spo spn plu plt spm ban mhy lac spk bam ama spj spi sph spg baf dps spd nph tdn spb spa chy bab baa btk chu tws bth tde rha bte cho xoo bta xom hhe twh son tcx gsu bsu tcr mgm noc ecv ecu tvo ecs mge ecp eco ecn cgl

mga ecj eci ech ece rfr cgb ecc tbr eca mfl rfe tbd smu daga nmu cfe reu ret

A.1.1 The Organism Selected by Phylogenetic Tree

Below are the shortened organism names which are selected by the phylogenetic tree method used for feature selection.

cfa rxy tan sma cyb cya tac dme cel ttj tth lpp cef lpn lpl vpa pho ilo lpf hdu pha cdi aha osa hwa pgi ago hch mca dkwa cch pfu cca pfo pfl mbo afu cbu pfa rba ayw pen lme tpv lma mtu hal mtp mac cal aeh ctr mth shm tpa cab mtc she blo cte llc lla cta msu hso pcu pcr ooe hsa csa xft dge sfr aci ace crp syw ljo xfa uur syn ava syg nfa syf sye syd abo syc pau pat neq par tma aba lil dsba det pai lic ptr pae pto mpu pac fal pab deh ftu mpn cpv cpt cps ftl aae sdn mpe fth cpn ftf mpa cpj sde swo pst psp cpa xcv ddi tko sco psb xcc xcb lga rno bga art vfi sus bfs bfr fra cne mmy mmu mmp mmo cmu san sak lxx sai mmc xac mma sag sto ppu hma stl cme ppr sth ste stc ppe dcin mle poy sso ehi ape dspd ldb ssc vvy vvu vch sru mka dpkn tfu lca bci fnu gvi lbu lbr lbl bbu mja lbj cjk dra pmu pmt ana pmn ter pmm efa pmi tel hit spz hin pma spy sps spr dsmi spn plt spm mhy lac spk spj spi sph spg baf spd nph spb spa chu tws bth tde rha cho xoo bta xom twh son tcx tcr noc ecu tvo mge cgl mga cgb tbr mfl smu cfe

A.1.2 The Organisms Selected by Fisher's Test

Below are the shortened organism names which are selected by the Fisher's Test method used for feature selection.

dspd dsmi ago dkwa neq osa dcgr dsba dkla ssc cal ddha pho sto sai tko mja ecu mka pab baf pfu mac ser mmp bbu afu bga crp spa ooe mge sso mma ape mga ljo lga sps mpu hpj pai twh sep ppe pto ctc ayw spm spi tel bci tvo tma cac spk spg sph spj mmo tws spy cfe spz ter uur mpn lbr sha spb hpy poy tde mtp mpe hpa hac tac rba fnu cpt lpl plt cca mfl bcc mhy cpa lac ani tpa det cta san blo ldb mmy lbu cab cpn ctr cte bas sak afm cpj stc cpe sag ava cyb cch lme cpf mth hwa cmu dsy fth tte pac stl cbu lca bca efa ste dyli pcu hma ftu aor ftf lmo bab lla cya pgi lwe chy xtr deh ssp sus chu nph lxx aba ftl tfu sab cpr buc wbr sco lpf swo cho gvi ana dncr dpyo lin art bcz rbe llc smu hal mle hhe gme sau sav cgb syn pfl aae lpp mxa baa lmf sfu bce dpkn bth bar lpn bpn wol bsu sar sao fra sam spn cdi cgl dge ade sas dar ehi bfl bqu mbo saa mtu bfs bfr cme bat ban tbr cjk bld pfa mtc bhe gox fal bli ama dde rfe cpv btk cps

dvu spd spr bcl psp sgl xfa pau syc dmgr pae dcnb gka oih cne vfi cef pmn sat pmt tpv sde