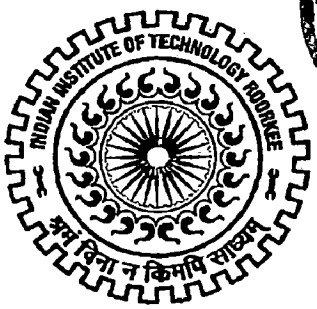
PREDICTING PROTEIN FUNCTION USING PHYLOGENETIC PROFILES

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

By

APPALA RAJU KOTARU



DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)
JUNE, 2009

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled **"PREDICTING PROTEIN FUNCTION USING PHYLOGENETIC PROFILES"** towards the partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science and Engineering** submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee (India) is an authentic record of my own work carried out during the period from August 2008 to June 2009, under the guidance of **Dr. R. C. Joshi, Professor, Department of Electronics and Computer Engineering, IIT Roorkee.**

The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute.

Date: 19.06.09
Place: Roorkee.

Appalaraju.K
(APPALA RAJU KOTARU)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 19.6.09
Place: Roorkee.

19/6
(Dr. R. C. JOSHI)
Professor
Department of Electronics and Computer Engineering
IIT Roorkee.

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my heartfelt gratitude to my guide **Dr. R. C. Joshi**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his invaluable advices, guidance, encouragement and for sharing his broad knowledge. His wisdom, knowledge and commitment to the highest standards inspired and motivated me. He has been very generous in providing the necessary resources to carry out my research. He is an inspiring teacher, a great advisor, and most importantly a nice person.

Thanks are due to Dr. Padam Kumar, Head of Institute Computer Center, Indian Institute of Technology Roorkee, for providing workstation in Unix Lab.

I also wish to thank Srikanth Isnaka, P Sundaramurthy and Shameer Kadhar for their valuable suggestions and timely help regarding the domain knowledge, and datasets. I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample moral supports and valuable suggestions.

On a personal note, I owe everything to the Almighty and my parents. The support which I enjoyed from my father, mother, elder and younger sister provided me the mental support I needed.

Appalaraju.k

APPALA RAJU KOTARU

Abstract

Predicting Protein Function is one of the important tasks of bioinformatics in post genomic era. Genome sequencing projects are scientific attempts that ultimately aim to determine the complete genome sequence of an organism. Although these sequences provide us with a lot of information, the functions of many of these are yet to be characterized. Computational biology methods provide powerful tools for this to minimize this sequence-function gap.

Though a large number of methods have been proposed and implemented for predicting protein function, a complete framework which considers all aspects for functional relatedness is missing. A great amount of research is carried out in finding the association based on similarity measures, constructing the phylogenetic tree and comparing them for phylogeny and assigning weights while finding the association, but still there are insufficient methods that have all the things.

In this Dissertation entitled "PREDICTING PROTEIN FUNCTION USING PHYLOGENETIC PROFILES", a solution is proposed which considers the co-evolution of the target genome which gives the basic similarity measure, the background phylogeny of reference genomes for profiles generation and assigning weights to the reference genomes. The ordering of genomes is used to show phylogeny which is computationally feasible.

The proposed strategy can be extended to increasing number of reference genomes. The accuracy of the predictions has been compared with existing approaches and the predictions are validated using the standard dataset. The possibility of using Functional Catalogue database for predicting protein function using Support Vector Machine classifier with radial basis as kernel function is also explored.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction and Statement of the Problem

1.1 Introduction

Predicting the functions of uncharacterized proteins from their sequence is a central goal of bioinformatics. The fully sequenced genomes of numerous organisms offer large amounts of information about cellular biology. It is a central challenge of bioinformatics to use this information in discovering the function of proteins [1]. Functional assignments of proteins come primarily from biochemical experimentation, which can be extended by matching recently sequenced proteins to those that have already been characterized. The problem of assigning functions to the remaining proteins is addressed here.

The huge amount of data that has accumulated over the years has made biological discovery via manual analysis tedious and cumbersome. This has in turn necessitated the use of techniques from the field of bioinformatics, an approach that is crucial in today's age of rapid generation and warehousing of biological data. Bioinformatics is the field of science in which biology, computer science and information technology merge to form a single discipline. The ultimate goal of the field is to enable the discovery of new biological insights as well as to create a global perspective from which unifying principles in biology can be discerned. At the beginning of the "genomic revolution", a bioinformatics concern was the creation and maintenance of a database to store biological information, such as nucleotide and amino acid sequences. Development of this type of database involved not only design issues but the development of complex interfaces where by researchers could both access existing data as well as submit new or revised data [2].

The computational methods for predicting protein function can provide an essential tool for the biologist, because many biological questions are directly answered when we understand the role of a protein in a biological process, how it interacts with other proteins and DNA and where in the cell it operates. Given the limitations of current

predictive methods, however, the purpose of such technology cannot be to replace experimentation, but rather to assist the biologist either by directly generating hypotheses to be verified experimentally or by suggesting a restricted set of candidate functions that can guide the exploration of promising hypotheses [3].

1.2 Motivation

Proteins are the most essential and versatile macromolecules of life and the knowledge of their functions is a crucial link in the development of new drugs, better crops and even the development of synthetic biochemicals such as biofuels.

Experimental procedures for protein function prediction are inherently low throughput because of huge experimental and human effort required in analyzing a single protein. Thus unable to annotate a non-trivial fraction of proteins that are becoming available due to rapid advances in genome sequencing technology. Release 40.3 of 26-May-2009 of UniProtKB(universal protein resource knowledge base)/TrEMBL(Translated European Molecular biology laboratory) [4] contains 7916844 sequence entries comprising 2577542687 amino acids. The growth of the database is summarized in the Figure 1.1.

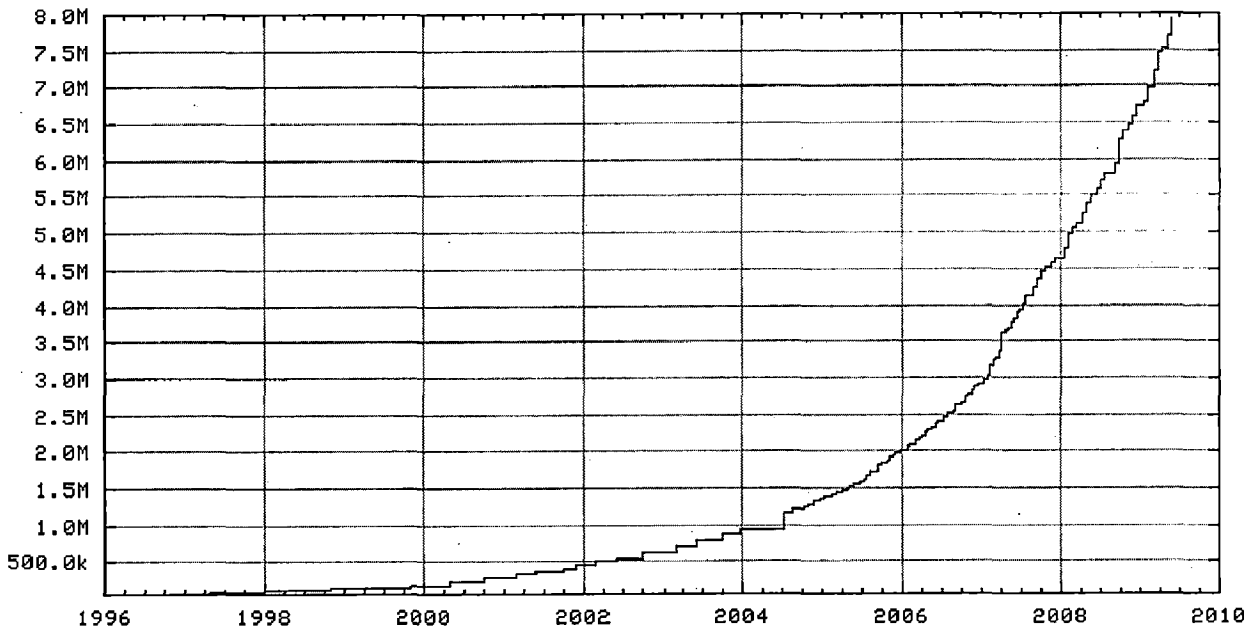


Figure 1.1 Number of Entries in UniProtKB/TrEMBL [4]

Currently, approximately 20%, 7%, 10% and 1% of annotated proteins in the Homo sapiens, Mus musculus, Drosophila melanogaster and Caenorhabditis elegans genomes, respectively, have been experimentally characterized (Annotations in Gene Ontology) [5]. This has resulted in a continually expanding sequence-function gap for the discovered proteins.

The state-of-the-art methods in text mining were presented in a competition for assessment of text mining systems in biology, the BioCreAtIvE [6] (Critical Assessment of Information Extraction systems in Biology) (BioCreAtIvE, 2006). One of the two "biologically meaningful" tasks defined by BioCreAtIvE was the automatic extraction of functional annotations to proteins from full-text documents related to them, by using the Gene Ontology (GO) classification system. Among the 20 participants, the best annotations were achieved with a perfect prediction percentage equal to 11.80%, which is still too low. This shows that automated methods for functional annotation of genes are still far from being perfect. And also the automated systems for the extraction of protein-protein interactions are performing with low accuracy rates.

These observations have motivated the development of computational techniques that utilize a high-throughput experimental data, phylogenetic profiles for protein function prediction with higher accuracies.

1.3 Statement of the Problem

The problem is to develop a technique for accurate and efficient way to predict the functions of a protein being queried from the database of proteins whose functions are known, using data mining techniques.

In this dissertation, we have made an attempt to design and implement the framework to solve the mentioned problem using two methodologies:

(i) Functional Protein Association Network

(ii) Functional Catalogue Database

1.4 Organization of the Report

This dissertation report comprises of six chapters including this chapter that introduces the topic and states the problem. The rest of the report is organized as follows.

Chapter 2 gives the background of protein functions and description of some well known prediction techniques in this field and Research Gaps.

Chapter 3 gives the design and implementation of the work done for the prediction of the protein function using the methodology, Functional Protein Association Network.

Chapter 4 gives the design and implementation of the work done for the prediction of the protein function using the methodology, Functional Catalogue Database.

Chapter 5 discusses the performance metrics used and the accuracy of the predictions has been compared with existing approached and the predictions are validated using the standard dataset.

Chapter 6 concludes the dissertation work and gives suggestions for future work.

Chapter 2

Background and Literature Review

2.1 Protein Function

The concept of protein function is highly context-sensitive and not very well-defined. In fact, this concept typically acts as an umbrella term for all types of activities that a protein is involved in, be it cellular, molecular or physiological.

Gene Ontology (GO) - The Gene Ontology Consortium's ontology GO [7], provides a dynamic controlled vocabulary for all organisms, with sufficient flexibility to accommodate the constant changes in biological knowledge. GO is aimed at providing a controlled terminology for labeling protein functions in a more precise, reliable and computer-readable manner.

GO maintains three separate taxonomies of terms, namely, "Molecular Function", "Biological Process" and "Cellular Component" as shown in Figure 2.1. Unlike other schemes, GO is not a tree-like hierarchy, but a Directed Acyclic Graph (DAG), where any term may have more than one parent as well as zero, one, or more children. This permits a more complete and realistic description of a term. Protein functions of any organism are described using the gene ontology.

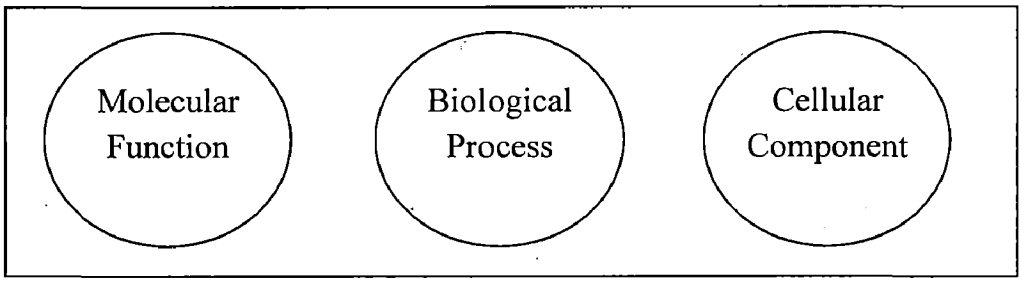


Figure 2.1 The three aspects of Gene Ontology Annotations

(a) **Molecular function**: Molecular function describes activities, such as catalytic or binding activities, that occur at the molecular level. GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the

actions and do not specify where or when, or in what context, the action takes place. Molecular functions generally correspond to activities that can be performed by individual gene products, but some activities are performed by assembled complexes of gene products.

(b) **Biological Process**: A biological process is series of events accomplished by one or more ordered assemblies of molecular functions.

(c) **Cellular Component**: A cellular component is just that, a component of a cell, but with the proviso that it is part of some larger object, this may be an anatomical structure or a gene product group.

The other definition is "function is everything that happens to or through a protein" [8]. The shape of a protein determines its biological activity. A single protein may have varying structure and more than one function. Proteins have many different biological functions. Proteins are classified according to their biological roles.

Enzymatic Proteins: The most varied and most highly specialized proteins are those with catalytic activity-the enzymes. Virtually all the chemical reactions of organic biomolecules in cells are catalyzed by enzymes. Many thousands of different enzymes, each capable of catalyzing a different kind of chemical reaction, have been discovered in different organisms. Digestive enzymes hydrolyze the polymers in food.

Transport Proteins: These proteins are involved in transporting other substances. For example, hemoglobin, the iron-containing protein of blood, transports oxygen from the lungs to other parts of the body. Other proteins transport molecules across cell membranes.

Structural Proteins: Structural proteins are very important for support. Collagen and elastin provide a fibrous framework in animal connective tissues, such as tendons and ligaments. Keratin is the protein of hair, horns, feathers, quills and other skin appendages of animals.

Storage Proteins: These proteins store amino acids. Ovalbumin is the protein of egg white, used as an amino acid source for the developing embryo. Casein, the protein of milk, is the major source of amino acids for baby mammals. Plants store proteins in seeds.

Hormonal Proteins: Hormonal proteins coordinate the bodily activities. Insulin, a hormone secreted by the pancreas, helps regulate the concentration of sugar in the blood.

Receptor Proteins: Receptor proteins are built into the membrane of a nerve cell and they detect chemical signals released by other nerve cells. They are involved in the cell's response to chemical stimuli.

Contractile Proteins: These proteins are very important in movement. Actin and myosin are responsible for the movement of muscles. Contractile proteins are responsible for the undulations of cilia and flagella, which propel many cells.

Defensive Proteins: These proteins protect against diseases. Antibodies combat bacteria and viruses.

2.2 Bioinformatics

The rate at which sequencing methods are producing genomic and proteomic data is far outpacing the rate at which these sequences are being experimentally annotated and understood. This trend is depicted in Figure 1.1. The number of human annotated proteins (Swiss-Prot, Protein Data Bank) is small compared to the number of proteins for which only the sequence is known (TrEMBL). In response, there has been a growing focus on ways to speed up the process of determining protein function through the use of computer systems that predict protein function.

2.2.1 Protein Function Prediction and Determination

In response to the overwhelming increase in protein sequence data, there has been much research in automated computational protein function prediction as demonstrated by the literature (Chapter 2.3) and the Automated Function Prediction Special Interest Group [9] meeting at the 2008 Intelligent Systems for Molecular Biology.

Protein function determination refers to the process of performing wet lab experiments to discover what function a protein serves. These methods can involve studying the protein's structure through Nuclear Magnetic Resonance or X-ray crystallography. Also, information about when proteins react or bind such as assays and 2-hybrid interactions are useful to understand the functions that a protein performs. Many approaches exist to understand what individual proteins do, however all of them are costly in terms of equipment and manpower.

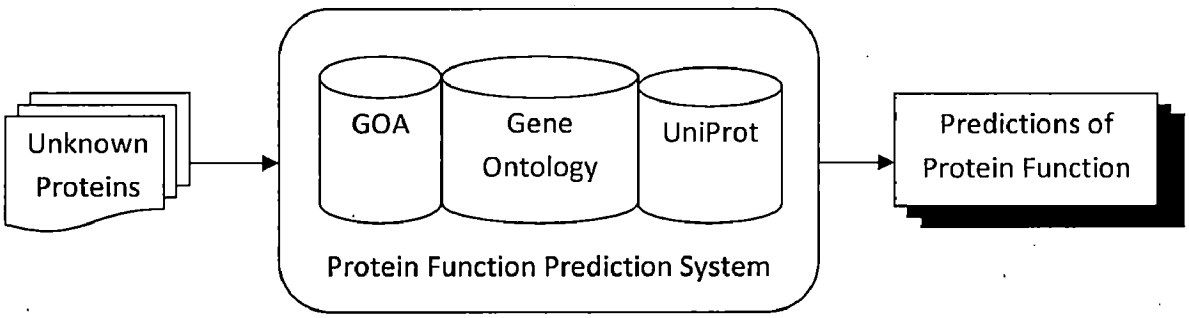


Figure 2.2 Protein Function Prediction

Protein function prediction provides biologists with predictions of the most likely functions that proteins perform as shown in Figure 2.2, which contains GOA (gene ontology annotations), Gene Ontology and UniProt(Universal protein resources) etc. This can help in the process of protein function determination by providing likely functions proteins perform and thus which experiments should be carried out. These methods should be highly accurate to be useful and they should be high-throughput so that they can be used for a large amount of data.

Another desirable feature of a prediction system is transparency [10]. Transparency refers to how well a user can understand why certain predictions were made. This can build a user's trust in the prediction system and thus can give clues as to the best experiments that should be performed in the wet lab. Alternatively, a user may decide that a prediction is incorrect by looking at the data used to make the prediction. Either way value is added when transparency is a feature of the prediction system.

Prediction methods often use machine learning approaches to model the problem domain. Machine learning leverages large datasets to extend knowledge about existing data and

supports the study of new, data which can speed up and increase the quality of, protein function determination.

2.2.2 Ontologies

In general, prediction is a mapping from instances to class. Before creating a prediction system, the type of predictions that it can make must be predefined. For example, in protein function prediction, we need to know what the possible protein functions are. Ontology is a set of terms describing the problem domain in a standardized way and defines the possible predictions that can be made. This addresses the issue of different researchers using different terminology to describe the same functions.

A variety of functions that proteins could perform are shown and various wet lab experiments could imply that a protein performs each of them.

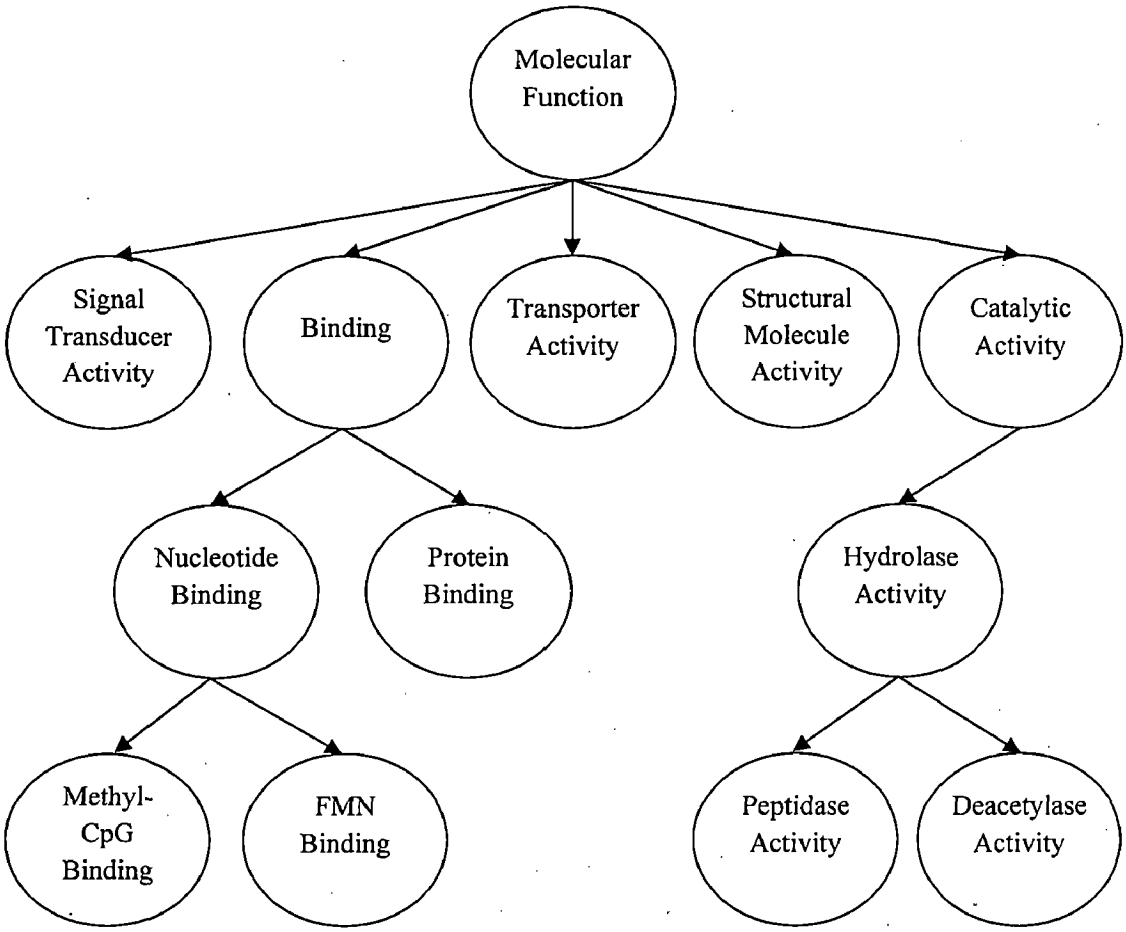


Figure 2.3 Hierarchical Ontology

Upon closer inspection it is evident that some functions are more similar to each other than others. For example, the functions "nucleotide binding" and "protein binding" are more similar to each other than either function is to "hydrolase activity". Furthermore, some functions are more general descriptions of the same function. For example, "peptidase activity" is a specific type of "hydrolase activity", in that every protein that performs the function "peptidase activity" necessarily performs the function "hydrolase activity". To represent these relationships between functions, the ontology can be structured in a hierarchy as shown in Figure 2.3. Hierarchically structured ontology such as the one shown in Figure 2.3 is called a hierarchical ontology.

2.3 Machine Learning

Machine Learning is an area of Artificial Intelligence that attempts to "learn" patterns and behaviors from real world data [11]. There are two major areas of machine learning: supervised and unsupervised learning.

In unsupervised learning, raw unlabeled data is given as input and the goal is to find patterns in this data. These patterns give information about similarities in the instances in the data set, but ultimately must be interpreted by users knowledgeable in the problem domain since no a priori knowledge about the data is given as input.

In supervised learning, the data given as input also includes associated labels with each instance in the data set. The labels are descriptions of the problem domain. The goal of supervised learning is to learn a function representing the data set, which can then be used to predict labels for future instances where the labels are unknown.

2.4 Classification

Classification is the process of learning a set of rules from instances (examples in a training set), or more generally speaking, creating a classifier that can be used to generalize from new instances [12]. The process of applying supervised Machine Learning to a real-world problem is described as shown in Figure 2.4.

The first step is collecting the dataset. A requisite expert could suggest which fields (attributes, features) are the most informative. The second step is the data preparation and data preprocessing. Depending on the circumstances, researchers have a number of methods to choose from to handle missing data. Instance selection is not only used to handle noise but to cope with the infeasibility of learning from very large datasets. Instance selection in these datasets is an optimization problem that attempts to maintain the mining quality while minimizing the sample size. It reduces data and enables a data mining algorithm to function and work effectively with very large datasets. Feature subset selection is the process of identifying and removing as many irrelevant and redundant features as possible. This reduces the dimensionality of the data and enables data mining algorithms to operate faster and more effectively.

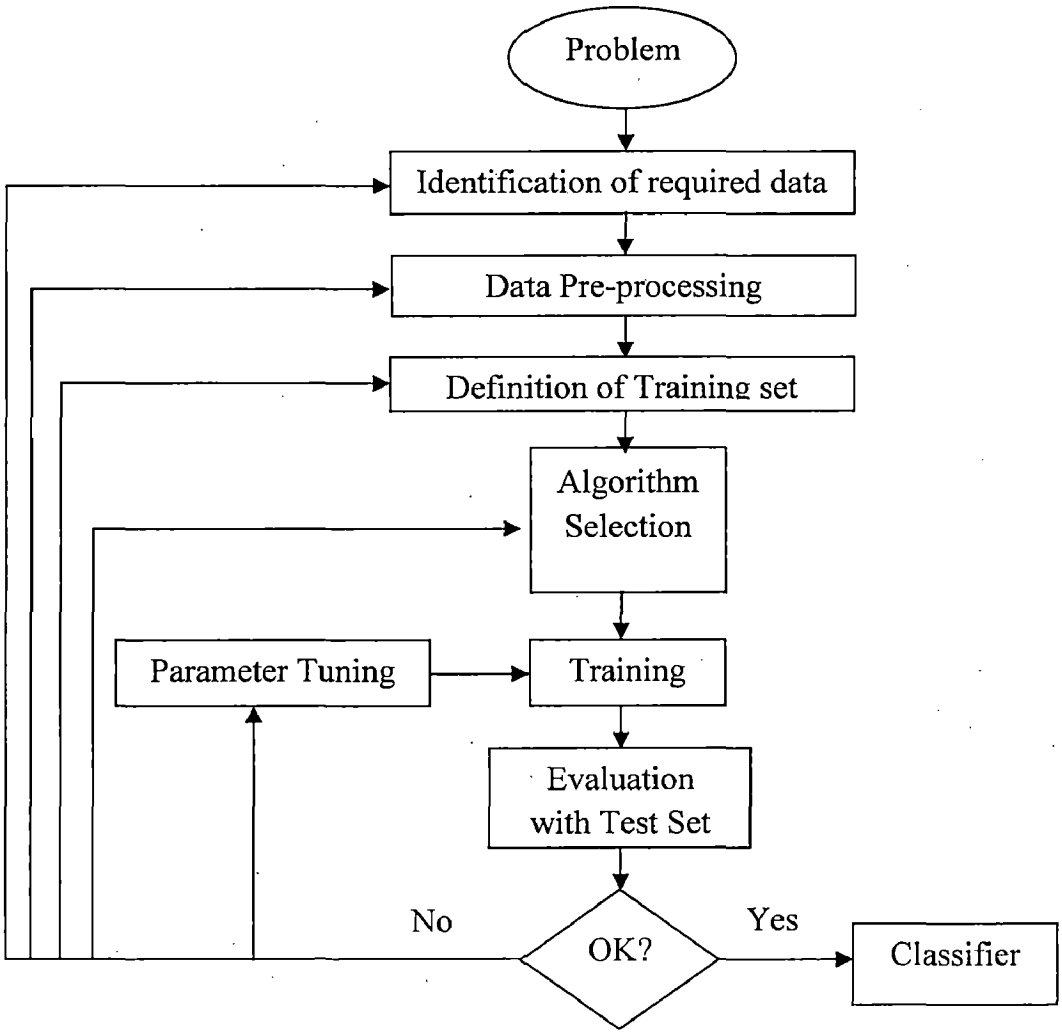


Figure 2.4 The process of Supervised Machine Learning

2.5 Literature Review

This section provides the various systems developed to predict the protein functions from protein database using various data mining techniques. There are three general classes of metrics that may be used to compare two binary phylogenetic profiles.

2.5.1 Methods Based on Co-Evolution

The first class of methods is insensitive to the underlying phylogeny of organisms and treats each position in the profile completely independent of the others.

The first study to analyze protein function using phylogenetic profiles was presented by Pellegrini et al. [1].

To represent the subset of organisms that contain a homolog, phylogenetic profile is constructed for each protein. This profile is a string with *n* entries, each one bit, where *n* corresponds to the number of genomes. The presence of a homolog to a given protein in the *n*th genome with an entry of unity at the *n*th position. If no homolog is found, the entry is zero. Proteins are clustered according to the similarity of their phylogenetic profiles. Similar profiles show a correlated pattern of inheritance and by implication, functional linkage.

This method predicts that the functions of uncharacterized proteins are likely to be similar to characterized proteins within a cluster. Phylogenetic profiles are computed for the 4,290 proteins encoded by the genome of *E. coli* by aligning each protein sequence (*Pi*) with the proteins from 16 other fully sequenced genomes. Proteins coded by the *n*th genome are defined as including a homolog of *Pi* if they align to *Pi* with a score that is deemed statistically significant.

This was a seminal study in this area and it opened the floodgates for protein function prediction using phylogenetic profiles.

Wu et al. [13] proposed a probability of matches using hypergeometric distribution. Let x and y be the number of lineages in which gene X and Y occur. Define the variable z as the number of lineages in which X and Y co-occur.

Chance co-occurrence probability distribution:
P (z|N, x, y) is the number of ways in which x and y can be distributed over N genomes, given that there are z co-occurrences, divided by the total number of ways x and y can be distributed without restriction as shown in Equation 2.1.

$$P = \frac{\omega_z \, \overline{\omega}_z}{W} \tag{2.1}$$

Where ω_z the number of ways to distribute z co-occurrences over the N lineages, $\overline{\omega}_z$ The number of ways of distributing the remaining $x - z$ and $y - z$ genes over the remaining N−z lineages, W is the number of ways of distributing X and Y over N lineages without restriction

It also advocated the use of more general measures of similarity for pairs of phylogenetic profiles. Three popularly used measures of similarity [14], namely the Hamming Distance (D), Pearson's Correlation Coefficient (r) and mutual information (MI) are evaluated for this task.
Hamming Distance (D)

$$D = x + y - 2z \tag{2.2}$$

Pearson Correlation Coefficient (r)

$$r = \frac{Nz - xy}{\sqrt{(Nx - x^2)(Ny - y^2)}} \tag{2.3}$$

Mutual Information (MI)

$$I(x,y) = \sum_{i,j} P_{i,j}(x,y) \, log_2 \frac{P_{i,j}\left(\frac{x}{y}\right)}{P_{i,j}(x)} \tag{2.4}$$

It is concluded from the analysis that, although the three measures are strongly related to each other, MI is the most informative measure of profile similarity for inferring functional relationship between two proteins.

This relationship is judged by membership of the proteins in the same metabolic pathway in KEGG (Kyoto Encyclopedia of Genes and Genomes) [15]. In addition, it is argued that proteins with complimentary profiles may suggest that they are functionally similar, which is likely to be missed if exact similarity of profiles is required.

However, these metrics do not consider the underlying phylogeny of the genomes in the profile. Accounting for phylogeny should improve our ability to detect truly co-evolving genes from that are merely present in a subset of related genomes.

Appala et al. [16] explored the feasibility of using supervised machine learning methods for predicting the protein function. Performance of traditional classification algorithms such as decision tree, naïve bayes and k-nearest neighbors were compared.

2.5.2 Methods Considering Underlying Phylogeny

The second class of metrics assumes that the underlying organism phylogenetic tree is known and takes advantage of this prior knowledge when computing profiles similarities. Vert [17] proposes the use of support vector machines (SVM) for learning protein functions from their phylogenetic profiles. However, instead of the common kernel functions used for SVMs, such as linear, a tree kernel is proposed to calculate the similarity of the profiles in the higher dimensional space used by SVM. This high-dimensional feature space is defined on the basis of the patterns of evolution of genes among the ancestors of the organisms under consideration, in a pre-specified phylogenetic tree. A linear time algorithm in the number of organisms, based on a post-order traversal of the tree, is also derived and its correctness proved.

The naïve kernel does not incorporate any knowledge about the nature of phylogenetic profiles, in particular the phylogenetic relationships among species. In order to create a

distance for phylogenetic profiles that reflects the similarity between evolutions they propose to map any profile to a feature space where each feature corresponds to a particular pattern of evolution. The tree kernel is the following, for any two profiles $(x_l , y_l) \square A^l * A^l$:

$$K(x_l , y_l) = \sum_{s \in c(T)} \sum_{z_s \in A^s} P(x_l | z_s)\ P(y_l | z_s) \tag{2.5}$$

Narra et al. [18] used the extended real-valued profiles to the above approach. Here, all the internal nodes of the phylogenetic tree are also assigned scores equal to the average of the scores at their children. An extended profile is now constructed for each protein by a post-order traversal of the tree. An SVM with a polynomial kernel is trained with these profiles and is used for function prediction. In evaluation using three-fold cross validation on the same data, performance better than that of Vert [17] is reported. The polynomial kernel function used is defines for vector x and y as:

$$K(x, y) = [1 + s\, D(x, y)]^d \tag{2.6}$$

where s and d are two adjustable parameters. Unlike ordinary polynomial kernel, D(x,y) is not the dot product of vector x and y, but rather, a generalized hamming distance for real value vectors.

Barker et al. [19] described a maximum likelihood statistical model for predicting functional gene linkages. This method detects independent instances of the correlated gain or loss of pairs of proteins on phylogenetic trees, reducing the high rates of false positives observed in conventional across-species methods that do not explicitly incorporate a phylogeny. It showed, in a dataset of 10,551 protein pairs, that the phylogenetic method improves by up to 35% on across-species analyses at identifying known functionally linked proteins.

This method showed that the protein pairs with at least two to three correlated events of gain or loss are almost certainly functionally linked. Contingent evolution, in which one gene's presence or absence depends upon the presence of another, can also be detected

phylogenetically and may identify genes whose functional significance depends upon its interaction with other genes. The improvement is derived from having a lower rate of false positives.

Zhou et al. [20] proposed a method based on evolutionary scenario which refers to a series of events that occurred in speciation over time, which can be reconstructed given a phylogenetic profile and a species tree. Common evolutionary pressures on two proteins can then be inferred by comparing their evolutionary scenarios, which is a direct indication of their functional linkage. This scenario method has proven to have better performance compared with the classical phylogenetic profile method, when applied to the same test set.

Barker et al. [21] proposed an approach that detects independent instances of the correlated gain and loss of pairs of genes from species genomes. It investigated the effect on results of basing evidence of correlations on two phylogenetic approaches, Dollo parsimony and maximum likelihood (ML). They further examined the effect of constraining the ML model by fixing the rate of gene gain at a low value, rather than estimating it from the data.

2.5.3 Methods Considering only Ordering

The third class of metrics is an approach that considers only an ordering of genomes and not a full phylogenetic tree.

Cokus et al. [22] proposed a method based on this kind of metric which considers only ordering of genomes. This paper shown that this approach is superior to the first class of metric which considered only co-evolution because the current method is considering both co-evolution and phylogeny. Scoring for the pair of genes is done using the below formula.

$$Score = log_{10}H - log_{10}R \tag{2.7}$$

Where H is the weighted hypergeometric p-value and R is weighted runs p-value for a pair of genes.

To test the performance of the proposed metric they computed the cumulative average $\log_{10}$ GO p-value. They restricted to the cellular components and biological process ontologies. The GO p-value is the probability that a randomly chosen benchmark pairs of genes has a common term atleast as specific as the most specific term common to the current pair of genes.

2.5.4 Other Methods

Certain amount of research is focused on selection of reference genome for construction of Phylogenetic Profiles. Sun et al. [23] suggested that reference organism should be selected based on genetic distance, rather than the relationship of taxonomy tree, because homology information used in the construction of phylogenetic profiles directly relies on the genetic distance of the sequences. And in paper by Loganantharaj et al. [24] selection of reference organism, for all members in a clade should evolve from a common ancestor and the one far apart from the rest is close to their ancestor. Therefore, select the organism that is evolutionarily the farthest apart from the rest of the organisms in that clade essentially selecting an outlier of that clade.

The approaches have been used in predicting function of some prokaryotic genomes quite successfully. The functional cohesiveness among clusters are week in eukaryotic target genomes, which is in contrast to some spectacular success in functional prediction in prokaryote [24]. It also suggested that Different mixture of reference sequences based on evolutionary history may help to improve the performance in function prediction in our target genome. Snitkin et al. [25] explored the application of phylogenetic profiling, method that explores the evolutionary co-occurrence of genes in the assignment of functional linkages, to eukaryotic genomes.

The measuring of approach's accuracy and coverage as well as to identify its biases, strengths and weaknesses is done by Raja et al. [26]. The conclusion it gave are selection

of genomes for reference set both at the super-kingdom level as well as within the eukaryotic kingdom affects the predictive power of this approach and adding a few eukaryote genomes into the reference set results in an improved performance. However, adding too many eukaryotes into the reference set decrease the performance. It also showed the null hypothesis, which involves comparison of the performances of the actual and the shuffled profiles, to assess the statistical significance of profile similarity.

A new technique, namely Annotating Genes with Positive Samples (AGPS), for defining negative samples in gene function prediction is proposed by Xing et al. [27]. The AGPS algorithm is different from existing methods, which have inappropriate assumptions about those genes that have no target annotations. Specifically, this approach do not simply regard those genes without target annotation as negative samples because one gene generally have multiple functions and it may indeed have the function even though it is not annotated with the target function currently.

2.6 Research Gaps

The first study to analyze protein function using phylogenetic profiles presented by Pellegrini et al. [1] considered only the similarity of the profiles and did not consider the background phylogeny of the genomes in the profile.

Probability of matches using hypergeometric distribution which is proposed by Wu et al. [13] did not incorporate the weights in the calculations and this also did not consider background phylogeny of the genomes in the profile. Factorial calculation is present in the methodology on the number of genomes and in the current data the number is 305 and its probability calculation is computationally expensive.

Yanum et al.[28] constructed weight-based profiles for reference proteins, but phylogeny is missing over here.

Tree kernel which is incorporated in support vector machine classifier proposed by Vert [17] is computationally expensive. There are more feasible kernel functions which were not explored.

Chapter 3

Protein Function Prediction Using Functional Protein Association Network

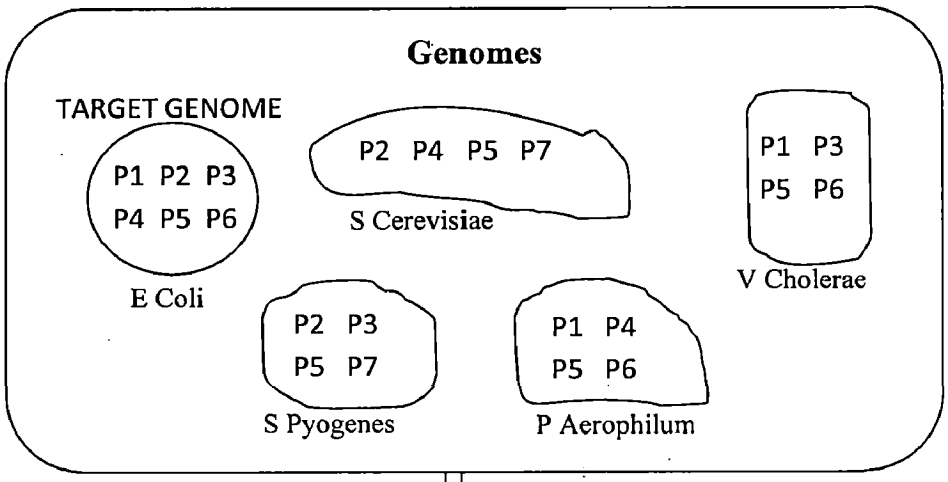
Predicting protein function using functional protein association network involves lot of Data processing techniques and Statistical Methods. Here, we have used a number of techniques to employ the system. These techniques range from wide areas of data mining such as hierarchal clustering and optimal leaf ordering for data processing and conditional probability for finding the top ranked interacting pairs.

3.1 Data Representation

The Phylogenetic profile of a protein can be described as a string that encodes the presence or absence of the protein from target genome in every sequenced reference genome. It is a binary vector whose length is the number of sequenced reference genomes. The vector contains 1 in the i^{th} position if the i^{th} genome contains a homologue of the corresponding gene, else a zero [1]. The homologue of the genes is obtained using BLASTP (protein-protein Basic Local Alignment Search Tool) [29] algorithm.

Some variations of these vectors use real numbers that reflect the extent of similarity between the original gene and the best match in the genome being searched, instead of 0s and 1s. Thus, these profiles provide a way of capturing the evolution of genes across various organisms. This information becomes useful for functional genomics when seen in the light of the phenomenon of speciation, which is the evolutionary mechanism by which new species are created from currently existing ones.

Phylogenetic profiles offer a very innovative method for inferring functional associations between proteins, since "functionally associated proteins are expected to have very similar phylogenetic profiles" [1]. This is the basic assumption made by all the approaches for function prediction on the basis of phylogenetic profiles. The generation of the profiles is shown in the Figure 3.1.



Phylogenetic Profiles:

EC	SC	SP	PA	VC
P1	0	0	1	1
P2	1	1	0	0
P3	0	1	0	1
P4	1	0	1	0
P5	1	1	1	1
P6	0	0	1	1

Figure 3.1 Phylogenetic Profiles Generation

3.2 Hypothesis of the Work

The first basic hypothesis is based on the similarity between the given two proteins. Greater the similarity more the proteins are functionally related [1].

The second hypothesis is based on the runs of consecutive matches both the proteins span. A run is defined as a maximal non-empty string of consecutive occupancy matches between two profiles. The profiles with more runs are more likely to involve functionally related proteins than profiles in which all the matches are concentrated in one interval of the tree [22]. For calculating runs the ordering of genomes is important and the procedure is explained in detail in the Section 3.3. The proof of the above hypothesis is showed by calculating the proposed methodology on the pairs and details are shown in results Section 5.1. Figure 3.2 shows this hypothesis.

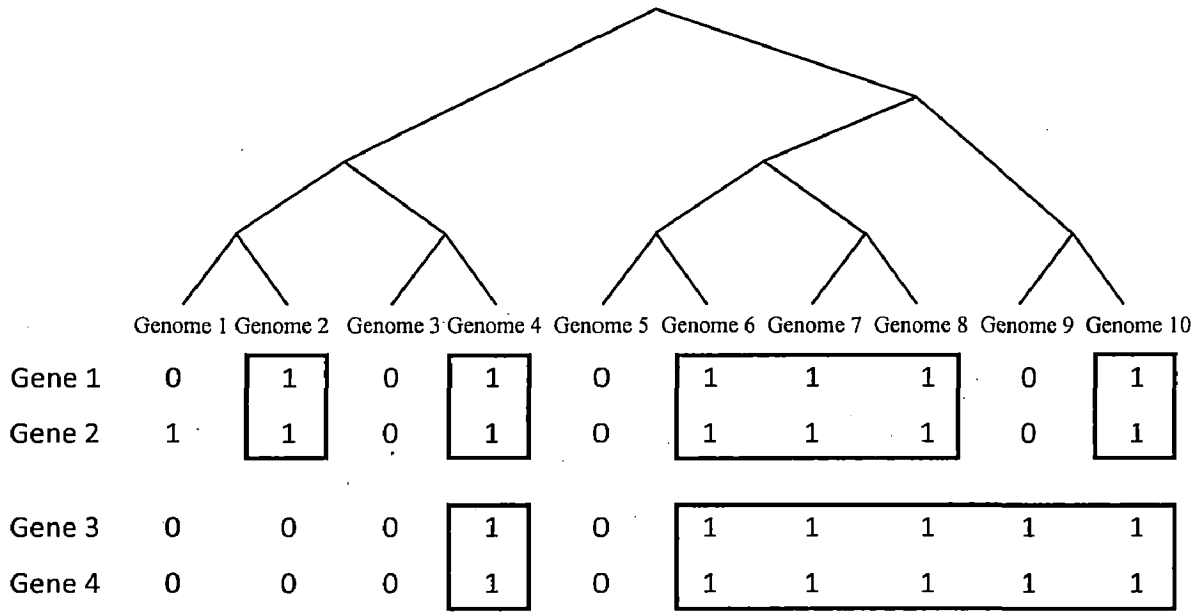


Figure 3.2 Phylogenetic Profiles showing the Runs Hypothesis.

Consider phylogenetic profiles for four genes. gene 1 and gene 2 have similarity six and in four runs while genes 3 and gene 4 also have similarity six and in two run. According to the second hypothesis we show that genes 1 and 2 are more likely to be truly co-evolving while genes 3 and 4 are likely to be just lineage-specific. Thus gene1 and gene 2 are more functionally related when compared to second pair and in the ranking of pairs the first pair gene 1 and gene 2 comes above the second pair gene3 and gene4.

3.3 Design of Proposed Methodology

The framework of our proposed automated protein function prediction system is as shown in Figure 3.3.

Separate components are provided in the framework for the following:

- Identifying the order of the genomes using the hierarchal clustering and optimal leaf ordering algorithm.
- Calculating the probability of the similarity between the given pairs.
- Calculating the probability of the runs between the given pairs.
- Finding the probability of the functional relatedness between the given pairs by calculating the total probability.

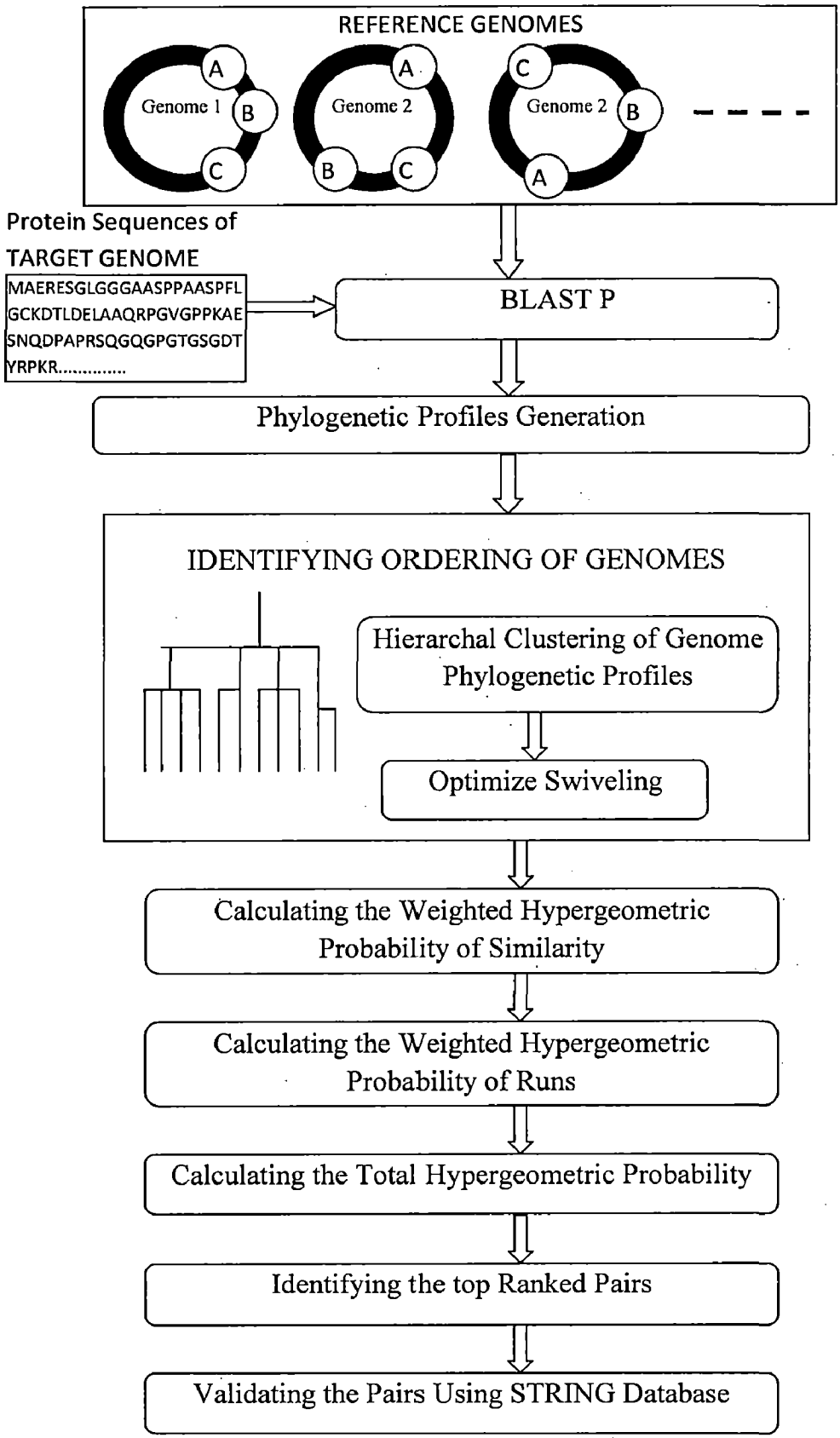


Figure 3.3 Design of Proposed Work

The Phylogenetic profile generation is explained in Section 3.1.

3.3.1 Hierarchal Clustering on Genomes Phylogenetic Profiles

The order of genomes is important because the number of runs generally changes as reference genomes are permuted. The ordering of genomes is established such that the order reflects the evolutionary relationships among the reference genomes [30]. Here, for hierarchal clustering, we used reference genomes phylogenetic profiles. Genome phylogenetic profiles are obtained as follows:

The phylogenetic profiles of the proteins of target genome consists of {0, 1} matrix whose rows are proteins and columns are the reference genomes. The genome phylogenetic profiles are the columns of the matrix.

The Procedure to perform hierarchical cluster is as shown in Figure 3.4.

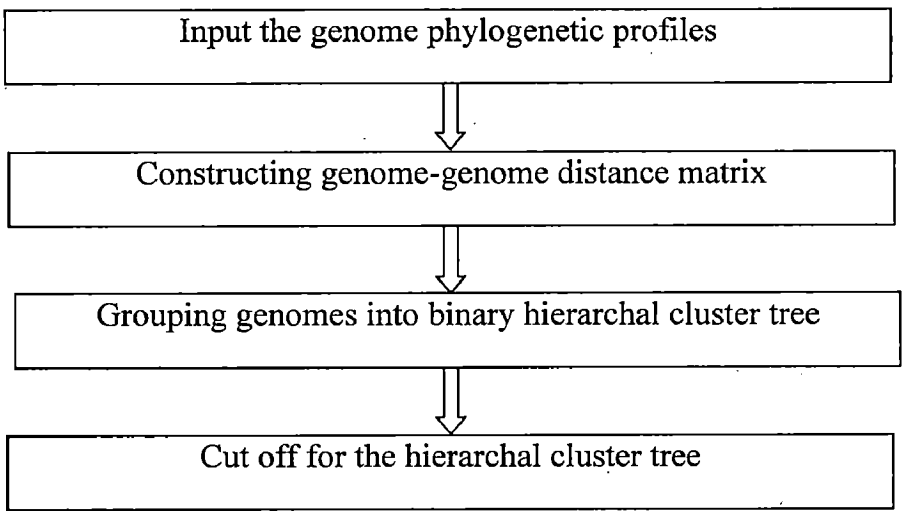


Figure 3.4 Hierarchal Clustering

a. Constructing Genome-Genome Distance Matrix:

For calculating the distance matrix, we used Jaccard dissimilarity to measure distance between two genomes, which is the percentage of disagreeing positions among positions where at least one gene has a 1. Jaccard dissimilarity formula is shown in Equation 3.1

Given an m-by-n data matrix X, which is treated as m (1-by-n) row vectors x_1, x_2, ..., x_m the Jaccard dissimilarity(d_{rs}) between the vector x_r and x_s are defined as follows:

$$d_{rs} = \frac{\#\left[(x_{rj} \neq x_{sj}) \wedge \left((x_{rj} \neq 0) \vee (x_{sj} \neq 0)\right)\right]}{\#\left[(x_{rj} \neq 0) \vee (x_{sj} \neq 0)\right]} \tag{3.1}$$

b. Grouping Genomes into Binary Hierarchal Cluster Tree

Genomes are paired into binary clusters, the newly formed clusters are grouped into larger clusters until a hierarchal tree is formed. Complete linkage is used here to form the pairs, also called furthest neighbor, uses the largest distance between objects in the two clusters.

n_r is the number of objects in cluster r. x_{ri} is the ith object in cluster r.

$$d(r,s) = \max\left(\text{dist}(x_{ri}, x_{sj})\right) \tag{3.2}$$

Where i $\epsilon(1,2,.....n_r)$ and j $\epsilon(1,2,.....n_s)$

c. Cut off for the Hierarchal Cluster Tree

Here, we need the ordering of the genomes which are leaves of the tree. So, we take the complete dendogram obtained in the above step

3.3.2 Optimal Leaf Ordering

The hierarchal clustering is only topological and there is an ambiguity about the ordering of genomes because of each non-leaf the left and right sub trees may be exchanged.

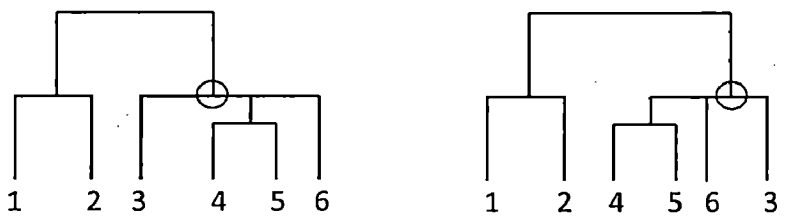


Figure 3.5 Effect of Node Flipping on the Leaf Ordering

An example of the effect of node flipping on the leaves ordering is shown in Figure 3.5. To optimize exchanges, we use the process of minimizing the sum of the Jaccard dissimilarities of pair wise adjacent genomes across the leaves of dendogram [31].

The process of to find the optimal swivellings is shown below:
For a tree T with n leaves, denote by $z_1,\ldots, z_n$ the leaves of T and by $v_1 \ldots v_{n-1}$ the n − 1 internal nodes of T. Since there are n-1 internal nodes, there are 2^{n-1} possible linear orderings of the leaves of a binary tree. To find an ordering of the tree leaves that maximizes the sum of the dissimilarities of adjacent leaves in the ordering. This could be stated mathematically in the following way. Denote by Φ the space of the 2^{n-1} possible orderings of the tree leaves. For $\varphi \in \Phi$, $D^{\varphi}(T)$ is defined as :

$$D\varphi(T) = \sum_{i=1}^{n-1} S\left(z_{\varphi_i}, z_{\varphi_{i+1}}\right)^2 \tag{3.3}$$

Where S(u,v) is the dissimilarity between two leaves of the tree. To find the ordering φ that minimize $D\varphi(T)$.

To find the optimal swivellings, Dynamic programming is used [22]. The left child of a node x is denoted by $l(x)$ and right child is denoted by $r(x)$. If x is a leaf then both $l(x)$ and $r(x)$ is x itself. Let $L(x)$ be the leaves of the subtree rooted at node x. For every $(x, \{a,d\})$ where x is a node and a is in $L(l(x))$ and d is in $L(r(x))$, keep track of the lowest cost $C(x, \{a,d\})$ among all swivellings of the subtree rooted at x that place a as the leftmost leaf and d as the rightmost leaf. Write $\Delta(b,c)$ for the additive cost for having leaf node b adjacent to leaf node c (which we took to be the square of their Jaccard dissimilarity). Then $C(x,\{x,x\}) = 0$ for every leaf x and the following recurrence relation for non-leaves x:

$$C(x,\{a,d\}) = min\{C(l(x),[a,b]) + \Delta(b,c) + C(r(x),[c,d])\} \tag{3.4}$$

b is L(l(l(x))) if a∈ L(r(l(x))) else b is L(r(l(x))),
c is L(l(r(x))) if d∈ L(l(r(x))) else c is L(r(r(x))).

Once the root, leftmost leaf and rightmost leaf are fixed, an optimal swivelling has to place some node b as the rightmost leaf of the left subtree and some node c as the leftmost leaf of the right subtree and use an optimal swivelling for each of these two subtrees. It is easy to compute all values of $C(x, \{-,-\})$ inductively on x from the bottom of the tree toward the root, finishing x for the left and right child of a node before beginning that node. The optimal cost for swivelling the whole tree is min(C(root, $\{a,d\}$)|a *in* L(l(root)) and d in L(r(root))).

3.3.3 Weighted Hypergeometric Similarity Probability

The weighted hypergeometric similarity probability is the probability of two profiles having a certain number of matches using an extension of the hypergeometric distribution that accounts for number of proteins in each genome. The basic assumption is that protein pairs with more matches in their profiles are more likely to co-evolve.

First, we calculate the weights w_i for values of i=1..n for each genome, which is the fraction in (0,1) of the 4195 reference genes contained in genes i. (For example, if a genome contains 75% of the genes in the reference genome, then its weight is 0.75). Genomes highly similar to the target genome have weights near to 1 while those more distant from it have lower weights. Weighted probability reduces to unweighted when all the weights are same. These weights are used in the calculation of both similarity probability and runs probability.

For n the number of genomes and consider a pair of genes, gene1 and gene2. For the similarity probability, the null hypothesis is genome i contains gene j are mutually independent over all pairs of genomes and genes.

The similarity probability for a pair of genes is that the number of genomes that have first gene in some number $a \geq 0$, the number of genomes that have the second gene is $b \geq 0$ and the number of genomes that have both genes in $c \geq 0$. The similarity p-value, then the number of genomes with both genes is at least as large as c given a and b is

$$P(c \geq \text{observed} \mid a, b) = \frac{P(c \geq \text{observed}, a, b)}{P(a, b)} \tag{3.5}$$

Let k take values 0,1…n and random variables A_k, B_k and C_k taking values in 0…k and A_k be the number of genomes that have the gene1, B_k be the number of genomes that have gene2 and C_k be the be the number of genomes that have both gene1 and gene2, restriction to genomes 0…k. To obtain conditional distribution of C_n given A_n and B_n it is sufficient to calculate the joint distribution A_n, B_n and C_n. Probability distributions are represented as multivariate polynomials with real coefficients in {0,1}. A multivariate polynomial is nothing more than an alternate representation of a multi-dimensional array of numbers. So if P` represent that table and here the variables are three, so it is a 3-dimensional table of each side n where the entries the possibility of occurrence of that combination. So the similarity p-value turns out to be as follows:

$$P(c \geq \text{observed} \mid a, b) = \frac{\sum_{c`=c}^{n} P`[a+1, b+1, c`+1]}{\sum_{c`=0}^{n} P`[a+1, b+1, c`+1]} \tag{3.6}$$

The details of calculation of the probability table P` is explained in implementation Section 3.6.

3.3.4 Weighted Runs Probability

The weighted hypergeometric runs probability is the probability of two profiles having a certain number of runs using an extension of the hypergeometric distribution that accounts for number of proteins in each genome. The basic assumption is that protein pairs with more runs in their profiles are more likely to co-evolve.

The runs probability for a pair of genes is that the number of runs that have first gene in some number $r \geq 0$, the number of runs that have in the second gene is $s \geq 0$ and the number of runs that have in both genes in $t \geq 0$ is the value of the unique entry of P that is P[r+1, s+1, t+1]. The runs p-value, then the number of genomes with both genes is at least as large as c given a and b is

$$P(t \geq \text{observed} \mid r, s) = \frac{P(t \geq \text{observed}, r, s)}{P(r, s)} \tag{3.7}$$

Let k take values 0,1…n and random variables R_k, S_k and T_k taking values in 0…k and R_k be the number of runs that have the gene1, S_k be the number of runs that have gene2 and T_k be the be the number of runs that have both gene1 and gene2, restriction to genomes 0…k. To obtain conditional distribution of T_n given R_n and S_n it is sufficient to calculate the joint distribution R_n, S_n and T_n. So if P`` represent that table and here the variables are three, so it is a 3-dimensional table of each side n where the entries the possibility of occurrence of that combination. So the runs p-value turns out to be as follows:

$$P(t \geq \text{observed} \mid r, s) = \frac{\sum_{t`=t}^{\frac{n}{2}} P`[r+1, s+1, t`+1]}{\sum_{t`=0}^{\frac{n}{2}} P`[r+1, s+1, t`+1]} \tag{3.8}$$

The details of calculation of the probability table P`` is explained in detail in implementation Section 3.6.

3.3.5 Total Probability

If H is the weighted hypergeometric similarity p-value for a given pair of genes and R is the modified weighted runs *p*-value for the same pair of genes, then we score the pair of genes as H*R or, on a logarithmic scale score is as follows.

$$Score = log_{10}H + log_{10}R \tag{3.9}$$

Lesser the score of a given pairs, more the pairs are functionally related.

3.4 Data Set

The phylogenetic profiles constructed from 305 genomes [32]. These profiles had been computed for each reference organism using BLASTP [29] to define the presence and absence of homologs across the genomes. Of all the 4,195 genes of the genome of Escherichia coli K12 are used as they have the most comprehensive annotations and therefore, allow us to more accurately assess the performance of methods. However,

there is no reason to expect that the results are specific to E. coli and therefore, expect the method to perform well if any of the fully sequenced genomes are used as target.

3.5 Collecting Benchmark Pairs

STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) is a database and web resource of known and predicted protein-protein interactions [33]. The STRING database contains information from numerous sources [34][35][36].

- Neighborhood
- Gene Fusion
- Co-occurrence
- Co-expression
- Experiments
- Databases
- Text Mining

These approaches are well known and widely accepted in case of prokaryotes, so it is always good to consider all of them. The data is weighted and integrated and a confidence score is calculated for all protein interactions. All the pairs whose score is greater than 0.5 are considered to be robust, so these pairs are considered for evaluation. The count of the number of pairs is 1,00,000 (1 lakh pairs).

3.6 Implementation Details of Proposed Methodology

This section presents the implementation details of the framework discussed in the earlier section. The individual modules in the previously discussed framework can perform independently from each other but in the same order as shown in the framework. The modules are implemented according to implementation convenience using different language tools like Matlab and Java. The implementation details are discussed below.

3.6.1 System Requirements

The programs are written in Matlab and Java. So, the system requires a Matlab software and Java Development Kit with Java Virtual Machine on the system.

Memory requirements for the hypergeometric probability calculation are as follows:
The multivariate variable in the probability calculation step of the framework is a three dimensional cube with each side of size equal to the number of genomes. We considered 305 genomes, so n is 305. The memory to hold a 3D cube of side 305 and datatype double is = (306 * 306 * 306) * 16 = 458441856 bytes, which is approximately 0.5 GB. We also used 4 temporary variables of this size. Total the memory required is approximately 3 GBytes RAM.

The system processor architecture requirements are as follows:
In 32-bit architecture, the maximum java virtual memory size is 1.5 GBytes. So, 64-bit architecture processor and 64 bit operating System are used.

3.6.2 Implementation of Genomes Ordering

The following Matlab functions are used.

- Y = pdist(X, 'jaccard') computes the Jaccard dissimilarity between pairs of objects in *n*-by-*p* data matrix X. Rows of X correspond to observations; columns correspond to variables. Y is a row vector of length $n(n-1)/2$, corresponding to pairs of observations in X. The distances are arranged in the order (2,1), (3,1), ..., (*n*,1), (3,2), ..., (*n*,2), ..., (*n*,*n*–1)).
- Dist = squareform (Y), where Y is a vector as created by the pdist function, converts y into a square, symmetric format Dist, in Dist Z(i,j) denotes the distance between the i^{th} and j^{th} objects in the original data.
- Tree = linkage(Dist , 'complete') creates a hierarchical cluster tree from the distances in Dist. Complete linkage, also called furthest neighbor, is used for the largest distance between objects in the two clusters.
- dendogram (Tree,0) generates a dendogram plot of the hierarchical, binary cluster tree represented by Z. Zero is given as second parameter to display the complete tree.
- Order = optimalleaforder (Tree, Dist) function is used to determine the optimal leaf ordering for the hierarchical binary cluster tree represented by Tree, using the distance matrix Dist.

3.6.3 Implementation of Probability Modules

Let n the number of genomes and consider a pair of genes. Let a_i, b_i and $c_i \in \{0, 1\}$ be 1 iff genome $i \in 1..n$ has the first gene, second gene and both genes, respectively. In the case of runs, it is useful to have a notional "0th genome" with $a_0 = b_0 = c_0 = 0$. Let r_i, s_i t_i $\in \{0, 1\}$ be 1 iff genome $i \in 1..n$ begins a run (i.e., genome i has a 1 and genome $i - 1$ has a 0) in the first and second genes and both genes respectively. With $r_0=s_0=t_0=0$ To determine whether a run starts in a given genome, the previous genome is also used. Weight w_i and variables A, B, C, R, S and T are define in Section 3.3.3, which takes cumulative values of a, b, c, r, s and t.

We divide the probability into four parts based on the values of a and b as shown in Figure 3.6

profile of the two genes at genome i	profile of the two genes at genome $i-1$			
	W $a_{i-1}b_{i-1} = 00$ $(c_{i-1} = 0)$	X $a_{i-1}b_{i-1} = 01$ $(c_{i-1} = 0)$	Y $a_{i-1}b_{i-1} = 10$ $(c_{i-1} = 0)$	Z $a_{i-1}b_{i-1} = 11$ $(c_{i-1} = 1)$
W $a_ib_i = 00$ $(c_{i-1} = 0)$ $(1-w_i)^2$	00 $r_i s_i t_i = 000$ 00 00 W_{i-1}	00 $r_i s_i t_i = 000$ 10 00 X_{i-1}	10 $r_i s_i t_i = 000$ 00 00 Y_{i-1}	10 $r_i s_i t_i = 000$ 10 10 Z_{i-1} $= W_i$
X $a_ib_i = 01$ $(c_{i-1} = 0)$ $(1-w_i)w_i b$	00 $r_i s_i t_i = 010$ 01 00 $s\, W_{i-1}$	00 $r_i s_i t_i = 000$ 11 00 X_{i-1}	10 $r_i s_i t_i = 010$ 01 00 $s\, Y_{i-1}$	10 $r_i s_i t_i = 000$ 11 10 Z_{i-1} $= X_i$
Y $a_ib_i = 10$ $(c_{i-1} = 0)$ $(1-w_i)w_i a$	01 $r_i s_i t_i = 100$ 00 00 $r\, W_{i-1}$	01 $r_i s_i t_i = 100$ 10 00 $r\, X_{i-1}$	11 $r_i s_i t_i = 000$ 00 00 Y_{i-1}	11 $r_i s_i t_i = 000$ 10 10 Z_{i-1} $= Y_i$
Z $a_ib_i = 11$ $(c_{i-1} = 1)$ $(w_i)^2abc$	01 $r_i s_i t_i = 111$ 01 01 $r\, s\, t\, W_{i-1}$	01 $r_i s_i t_i = 101$ 11 01 $r\, t\, X_{i-1}$	11 $r_i s_i t_i = 011$ 01 01 $s\, t\, Y_{i-1}$	11 $r_i s_i t_i = 000$ 11 11 Z_{i-1} $= Z_i$

Figure 3.6 Division of the Probability Calculation

$$P_k = W_k + X_k + Y_k + Z_k \tag{3.10}$$

W_k represent the possibility of a=0 and b=0

X_k represent the possibility of a=0 and b=1

Y_k represent the possibility of a=1 and b=0

Z_k represent the possibility of a=1 and b=1

And initial values are $W_k = 1$, $X_k = 0$, $Y_k = 0$, $Z_k = 0$

(a) Similarity Probability calculation:

$$W_o^{`} = 1$$

$$X_o^{`} = Y_o^{`} = Z_o^{`} = 0$$

$$W_i^{`} = (1 - w_i)^2 \left( W_{i-1}^{`} + X_{i-1}^{`} + Y_{i-1}^{`} + Z_{i-1}^{`} \right)$$

$$X_i^{`} = (1 - w_i) w_i b \left( W_{i-1}^{`} + X_{i-1}^{`} + Y_{i-1}^{`} + Z_{i-1}^{`} \right)$$

$$Y_i^{`} = (1 - w_i) w_i a \left( W_{i-1}^{`} + X_{i-1}^{`} + Y_{i-1}^{`} + Z_{i-1}^{`} \right)$$

$$Z_i^{`} = w_i^2 abc \left( W_{i-1}^{`} + X_{i-1}^{`} + Y_{i-1}^{`} + Z_{i-1}^{`} \right)$$

$$P_i^{`} = \left( W_i^{`} + X_i^{`} + Y_i^{`} + Z_i^{`} \right)$$

Cubical array is used to store a polynomial in a, b, c with increasing successive powers 0, 1, 2, . . . of 'a' going front-to-back, of 'b' going top-to-bottom and of 'c' going left-to-right. We start with an array consisting of a single element +1.0, i.e., $P_0^{`}$.

For $i \in 1..n$, replace the array with the entry wise sum of following four arrays (corresponding to the four terms of the first factor of the right-hand side of $P`_i$):

(1) The current array with every entry multiplied by $(1 - w_i)^2$ and padded by a 1-entry-thick slab of +0.0's on the back, bottom and right.

(2) The current array with every entry multiplied by $(1 - w_i)w_i$ and padded by a 1-entry-thick slab of +0.0's on the back, top and right.

(3) The current array with every entry multiplied by $w_i(1-w_i)$ and padded by a 1-entry-thick slab of +0.0's on the front, bottom and right.

(4) The current array with every entry multiplied by $(w_i)^2$ and padded by a 1-entry-thick slab of +0.0's on the front, top and left.

The final array gives P_n`` and is easily post-processed in a single last pass to obtain a full set of the desired p-values for the given pairs

(b) Runs Probability

$$W_i^{\cdots} = (1 - w_i)^2 \left(W_{i-1}^{\cdots} + X_{i-1}^{\cdots} + Y_{i-1}^{\cdots} + Z_{i-1}^{\cdots} \right)$$
$$X_i^{\cdots} = (1 - w_i) w_i \left(sW_{i-1}^{\cdots} + X_{i-1}^{\cdots} + sY_{i-1}^{\cdots} + Z_{i-1}^{\cdots} \right)$$
$$Y_i^{\cdots} = (1 - w_i) w_i \left(rW_{i-1}^{\cdots} + rX_{i-1}^{\cdots} + Y_{i-1}^{\cdots} + Z_{i-1}^{\cdots} \right)$$
$$Z_i^{\cdots} = w_i^2 \left(rst\, W_{i-1}^{\cdots} + rtX_{i-1}^{\cdots} + stY_{i-1}^{\cdots} + Z_{i-1}^{\cdots} \right)$$
$$P_i^{\cdots} = \left(W_i^{\cdots} + X_i^{\cdots} + Y_i^{\cdots} + Z_i^{\cdots} \right)$$

Cubical array is used to store a polynomial in r, s, t with increasing successive powers 0, 1, 2, . . . of 'r' going front-to-back, of 's' going top-to-bottom and of 't' going left-to-right. We start with four arrays W, X, Y and Z.

For i ∈ 1...n, simultaneously update W, X, Y and Z as follows:

(1) Replace W with entry wise sum of four current arrays after multiplying each element by $(1\text{-}w_i)^2$ and padding by a 1-entry-thick slab of +0.0's on the back, bottom and right.

(2) Replace X by following : take entry wise sum of current W and Y after padding by a 1-entry-thick slab of +0.0's on the back, top and right, and current X and Z after padding by a 1-entry-thick slab of +0.0's on the back, bottom and right, then multiply by $(w_i)^2$.

(3) Replace Y by following : take entry wise sum of current W and X after padding by a 1-entry-thick slab of +0.0's on the front, bottom and right and current Y and Z after padding by a 1-entry-thick slab of +0.0's on the back, bottom and right, then multiply by $(1 - w_i)\, w_i$.

(4) Replace Z by following : take entry wise sum of current W after padding by a 1-entry-thick slab of +0.0's on the front, top and left, and current X after padding by a 1-entry-thick slab of +0.0's on the front, bottom and left and current Y after padding by a 1-entry-thick slab of +0.0's on the back, top and left and current Z after padding by a 1-entry-thick slab of +0.0's on back, bottom and right.

The final array gives P_n`` is obtained by taking entry wise sum of the final W, X, Y and Z arrays. Now $P_n$`` is easily post-processed in a single last pass to obtain a full set of the desired p-values for the given pairs.

Chapter 4

Protein Function Prediction Using Functional Catalogue Database

Predicting protein function using Functional Catalogue Database involves lot of Data processing techniques and computational methods. Here, we have used Classification techniques to employ the system. The basic architecture of two stage supervised learning used is shown in Figure 4.1.

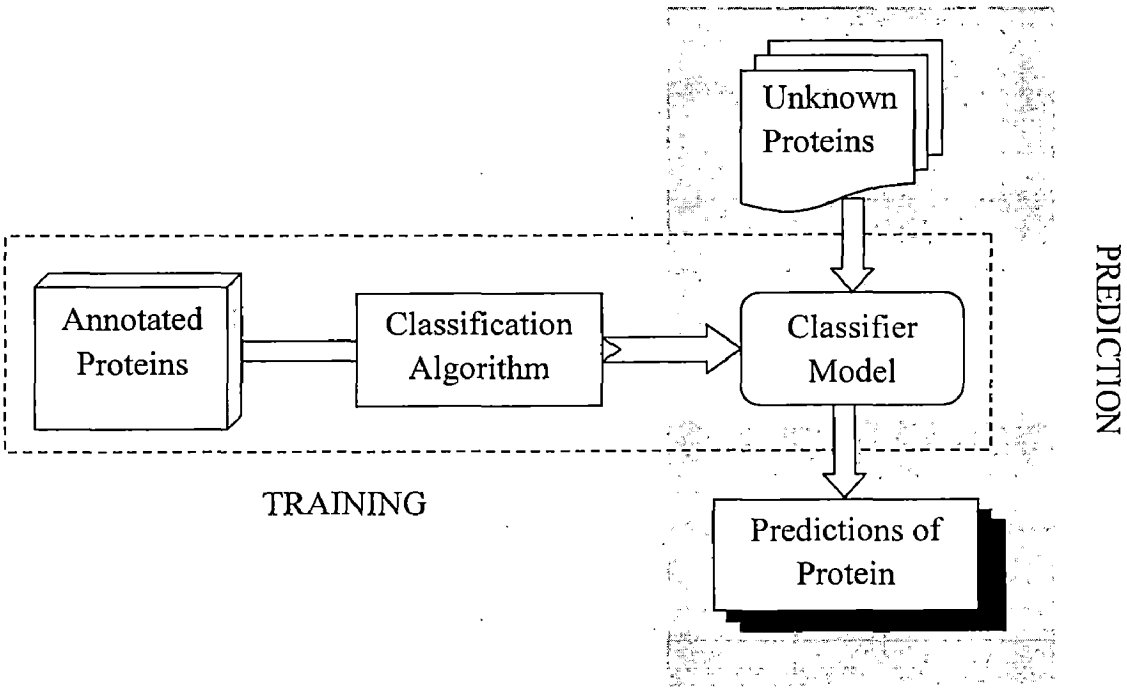


Figure 4.1 Two stages of Supervised Learning

4.1 Design of Proposed Methodology

The framework of our proposed automated protein function prediction system is as shown in Figure 4.2.

Phylogenetic profiles generation is explained in the Section 3.1. Class labeling is the process of determining the functional classes of the different protein and it is very much ongoing process and to a large extent one of the key steps in understanding the genomes

of the various species. We considered yeast genomes as target genomes and fortunately, in the case of the yeast genome, there exist extensive annotations for a large fraction of the genes. For our study, we used the functional annotations that are available in the MIPS(Munich Information Center for Protein Sequences) database [37].

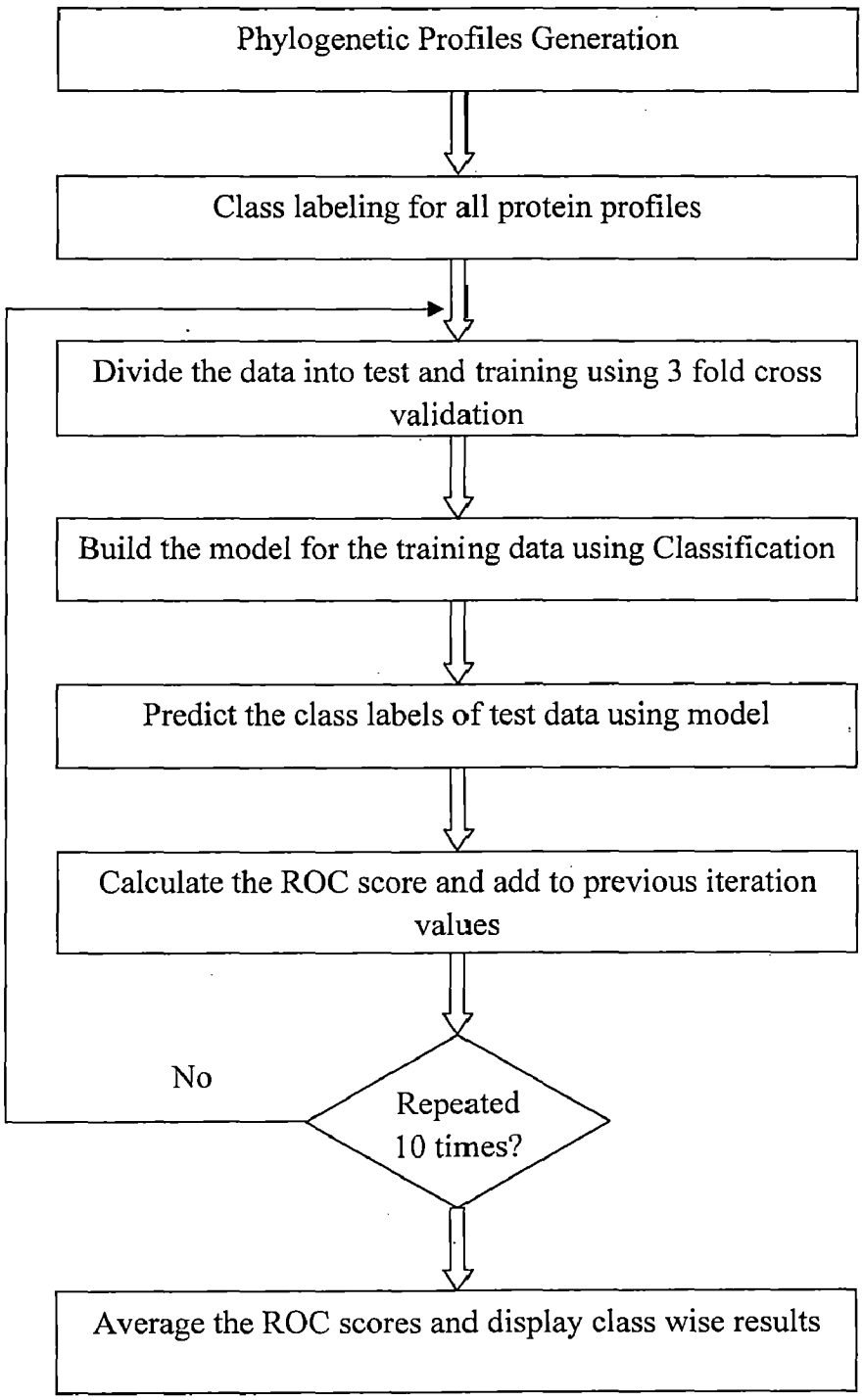
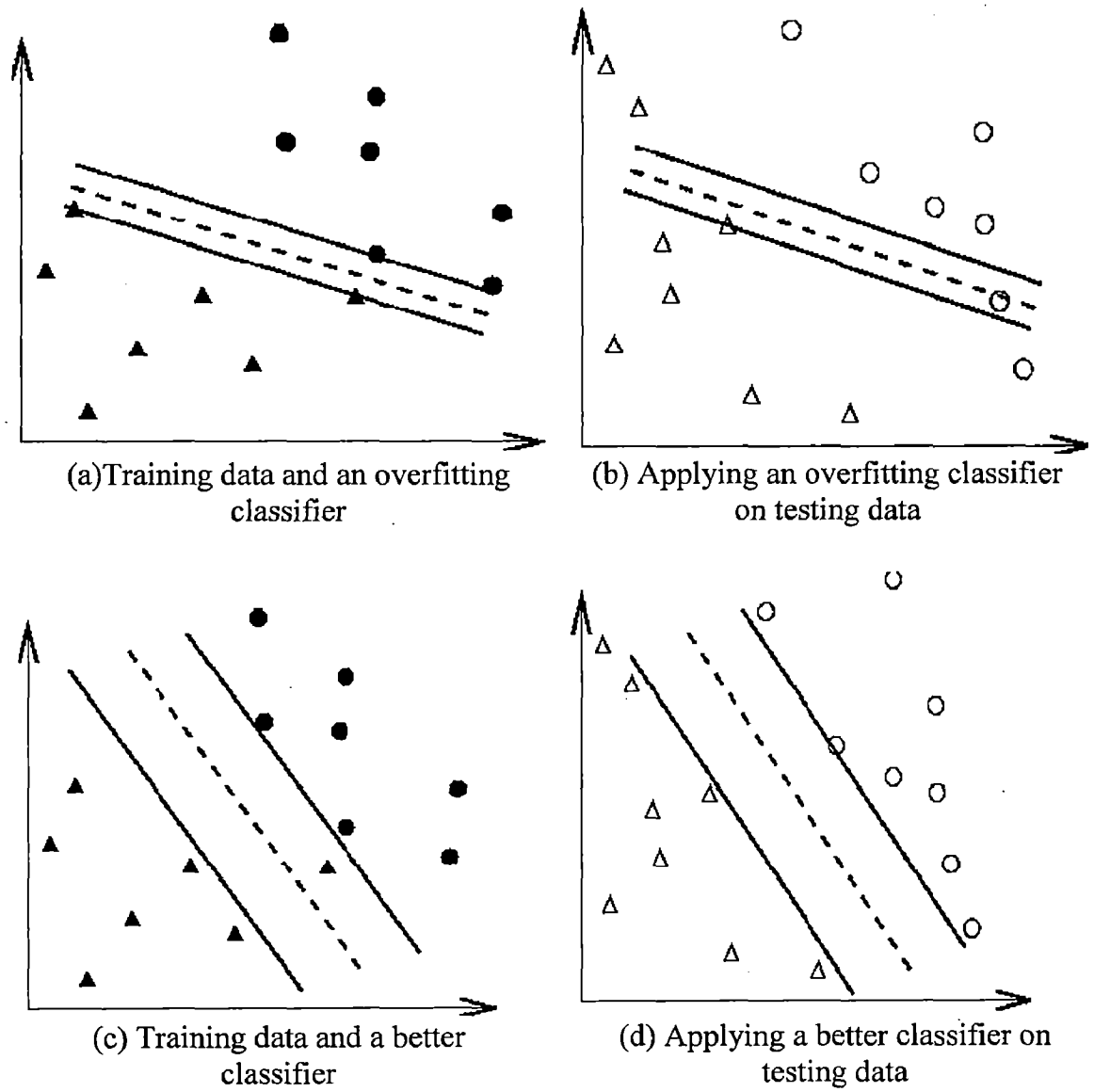


Figure 4.2 Design of Proposed Methodology

4.1.1 Cross Validation

Cross validation is used to minimize the empirical error, i.e. the error made on the data used to train the algorithm. A current pitfall when we try to do this is to stick too closely to the data : we learn irrelevant details of the training set, which leads to a wrong generalization. This problem happens when we have too little data and a too precise model.



(a)Training data and an overfitting classifier

(b) Applying an overfitting classifier on testing data

(c) Training data and a better classifier

(d) Applying a better classifier on testing data

Figure 4.3 An Overfitting Classifier and a Better Classifier (Dark circle and triangle: training data; Hollow circle and triangle: testing data)

In v-fold cross-validation, we first divide the training set into v subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining v-1 subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified.

The cross-validation procedure can prevent the overfitting problem. We use Figure 4.3 which is a binary classification problem (triangles and circles) to illustrate this issue. Filled circles and triangles are the training data while hollow circles and triangles are the testing data. The testing accuracy the classifier in Figures 4.3(a) and 4.3(b) is not good since it overfits the training data. If we think training and testing data in Figure 4.3(a) and 4.3(b) as the training and validation sets in cross-validation, the accuracy is not good. On the other hand, classifier in Figure 4.3(c) and 4.3(d) without overfitting training data gives better cross-validation as well as testing accuracy.

4.1.2 Classification

Over the years a variety of different classification algorithms have been developed by the machine learning community. Depending on the characteristics of the data sets being classified certain algorithms tend to perform better than others. In recent years, algorithms based on support vector machine have been shown to produce reasonably good results for problems in which the independent variables are homogeneous. For this reason, we primarily used this classification algorithm.

Support Vector Machine: Support Vector Machine (SVM) is a learning algorithm proposed by Vapnik [38]. This algorithm is introduced to solve two-class pattern recognition problems. Given a training set in a vector space, this method finds the best decision hyper plane that separates two classes. The quality of a decision hyper plane is determined by the distance (referred as margin) between two hyper planes that are parallel to the decision hyper plane and touch the closest data points of each class. The best decision hyper plane is the one with the maximum margin. By defining the hyper plane in this fashion, SVM is able to generalize to unseen instances quite effectively. The SVM problem can be solved using quadratic programming techniques [39]. SVM extends

its applicability on the linearly non-separable data sets by either using soft margin hyper planes, or by mapping the original data vectors into a higher dimensional space in which the data points are linearly separable. The mapping to higher dimensional spaces is done using appropriate kernel functions, resulting in efficient algorithms. A new test object is classified by looking on which side of the separating hyper plane it falls and how far away it is from it.

In their basic form, SVMs learn linear decision rules $h(\vec{x}) = sign(\vec{w}.\vec{x} + b)$ described by a weight vector $\vec{w}$ and a threshold b. Let the input be a sample of n training examples with the j^{th} input point being $x^j = (x_1^j, x_2^j, ..., x_n^j)$.

Let this input point be labeled by the random variable $Y^j \in \{-1, +1\}$. For a linearly separable input, the SVM finds the hyperplane with maximum Euclidean distance to the closest training examples. This distance is called the margin δ as depicted in Figure 4.4 For non separable training sets, the amount of training error is measured using slack variable ξ^j as shown in Figure 4.4 for a two class problem. Computing hyperplanes is equivalent to solving the following primal optimization problem.

minimize

$$V(\vec{w}, \vec{b}, \vec{\xi}) = \frac{1}{2}\vec{w}.\vec{w} + C\sum_{i=1}^{n}\xi^j \tag{4.1}$$

subject to

$$\forall_{j=1}^{n} : y^j[\vec{w}.\vec{x}^j + b] \geq 1 - \xi^j \tag{4.2}$$

$$\forall_{j=1}^{n} : \xi^j > 0 \tag{4.3}$$

The second constraint requires that all the training examples are classified properly up to a slack ξ. Therefore, $\sum_{j=1}^{n}\xi^j$ is an upper bound on the number of training errors. The factor C in Equation 4.1 is a parameter that allows trading off training error verses model complexity. Note that the margin of the resulting hyperplane is $\delta = 1/\|\vec{w}\|$. The hyperplane

that separates the positive from the negative examples and has maximal margin is called the maximal margin hyperplane or the Optimal Separating Hyperplane (OSH) as shown in Figure 4.4. The hyperplanes that contain the training points with the minimal distance to the OSH are called the margin hyperplanes and they form the boundary of the margin. They are represented as *H*1 and *H*2 in Figure 4.4.

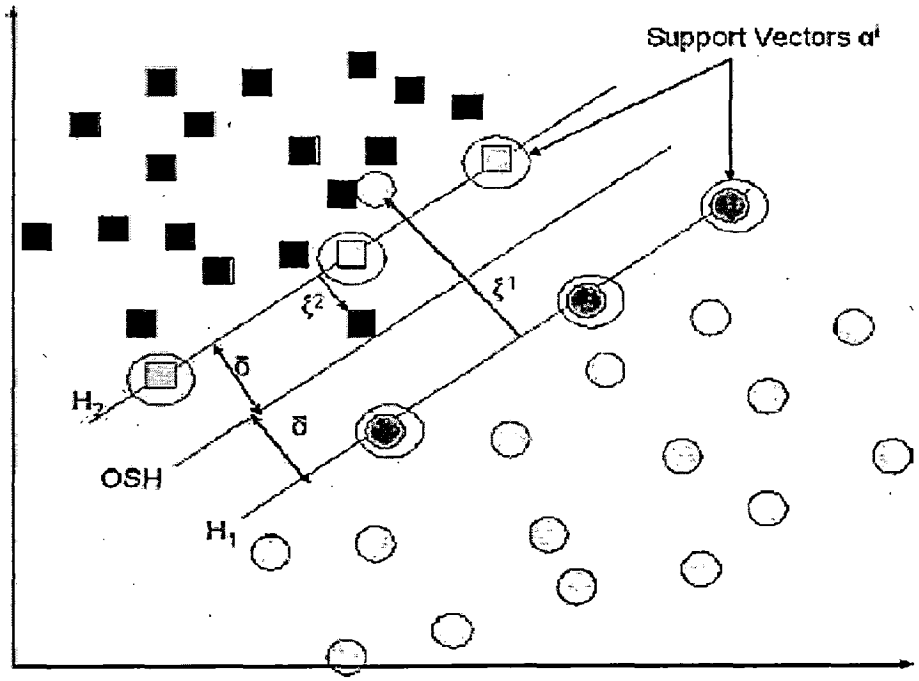


Figure 4.4 The Optimal Separating Hyperplane (OSH), Support Vectors α_i and the Slack Variables ξ_i

4.1.3 Kernel Function

The performance of SVM classification is strongly related to the choice of the kernel function and the penalty parameter C. There are a large number of kernel functions available. The RBF(Radial Basis Function) kernel non-linearly maps samples into a higher dimensional space and can handle the case when the relation between class labels and attributes is nonlinear. And when the number of instances is much greater than the number of features then, non linear kernel is used. So RBF kernel is used. The RBF kernel can be described for vectors a and b as follows

$$K(a,b) = e^{-gamma* \|a-b\|^2} \tag{4.4}$$

For finding the optimum values of parameters (C, γ) automatically, a grid search technique is used using cross validation. And for this kernel function, we tuned the classifier and got the gamma value as 1 and C value as 1. The kernel functions are conveniently implemented in the open source software package SVM Light [39] which is used in this work.

4 .2 Data Set

We used the yeast Saccharomyces cervisiae genome the same dataset as in [17], [18]. Proteins with accurate functional classification were selected. The phylogenetic profiles of 2465 yeast genes selected for their accurate functional classifications were generated for each target organism using BLASTP [29] to define the presence and absence of homologs across the genomes.

For the data, there are 251 gene functional classes organized in tree structure. Based on the amount of information that is known for each gene, the MIPS database [37] assigns it to one or more nodes of the functional classes. Genes for whom detailed functional information is known tend to be assigned towards the leaves of the tree, whereas genes for which the information is more limited tend to be assigned at the higher-level nodes of the tree. For example, a gene YHR037W is assigned a function named amino-acid biosynthesis. Because amino-acid biosynthesis which is also a sub-function of the top-level function metabolism, YHR037W has all those functions, {amino-acid biosynthesis, amino-acid metabolism, metabolism}, a function at a node and all the functions of its path to the top-level node. A gene also may have functions assigned from multiple branches. For the case of YAL001C it has functions from the top level classes transcription, cellular organization and their subcategories. As a result of this functional class assignment, each gene has 3.4 functions assigned on the average. The distribution of the number of classes at the different level of the tree is shown in Table 4.1.

Most of the functions are small in their size, which makes functionality prediction difficult. For this reason only functional classes that contain at least 10 genes were extracted which resulted in 133 functional classes.

Table 4.1 Number of defined function categories at each level in the tree.

Level	Functions
1	16
2	107
3	86
4	40
5	2

4.3 Implementation Details of Methodology

SVM is trained for each functional category to predict whether a gene should be assigned to it or not based on phylogenetic profiles. A 3-fold cross validation was adopted for the experiments. For each functional class, two third of members are randomly selected as positive training examples and rest as positive testing examples. Genes not belonging in that class were randomly split into two thirds as negative training and one third as negative testing examples. Now, the positive training examples and negative training examples are combined to form training data and positive testing examples and negative testing examples are combines to form testing data for that particular class. After generating the data model build using training data and the model is tested using the test data. The performance of the classifier is measured using the ROC scores. ROC curve plots true positive rate (TPR) on y-axis and false positive rate (FPR) on x-axis.

$$TPR = \frac{TP}{TP + FN} \tag{4.5}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.6}$$

Where TP is true positive, FN is false negatives, FP is false positive, TN is true negatives. Here we plotted true positives as a function of false positives. After training the SVM with training data the output file we get when we test the model with test data contains a score related to the distance between the test example and the linear boundary in the feature space. As each functional class contains a small number of genes learning problem is very unbalanced (there are few positive examples but many negative ones).

This issue is handled by giving more weight to the positive examples in SVM learning. Moreover, this implies that only a small percentage of false positives can be tolerated in real world applications (such as function predictions), so we measured the ROC50 score for each SVM, i.e., the area under the ROC curve up to the first 50 false positives [40].

Chapter 5

Results and Discussions

5.1 Results of Protein Association Network Method

The phylogenetic profiles generated consider 305 genomes. As mentioned in the Section 3.5, all the pairs whose score is greater than 0.5 are considered for evaluation. The total number of pairs are 1,00,000 (1 lakh). The weighted hypergeometric probability with runs proposed by Cokus et al. [22] outperformed all the existing methods which uses the following methodology:

a. Unweighted hypergeometric: This method is based assumption that all the genomes in the profiles generation are given equal weightage. And the phylogenetic relation which is present at the background is not considered. Drawback of this is it cannot be implemented for our method because it contains the factorial calculation which cannot be calculated for the updated data which contains string of length 305.
b. Mutual information: This method gave the measure as the entropy of first profile plus the entropy on second profile minus the entropy of the joint profile viewed one genome at a time. Drawback of this case is also the phylogenetic relation is not considered.
c. Weighted hypergeometric: This method gave the weightage to the genome by which the profiles are generated which is not considered in the above methods. But the drawback of this method is the phylogenetic relation is not considered.

So, we compared our method with weighted hypergeometric probability with runs, which implies that if the results are better than this method then its obvious that it is better than all the above methods.

The drawbacks of the methods are mentioned in detail in the literature review section. We also compared our result with weighted hypergeometric method. All the three methods are benchmarked against pairs which are obtained from the STRING [36] database.

5.1.1 Comparison Using Benchmark Pairs

Figure 5.1 compares three methods, considering one method at a time.

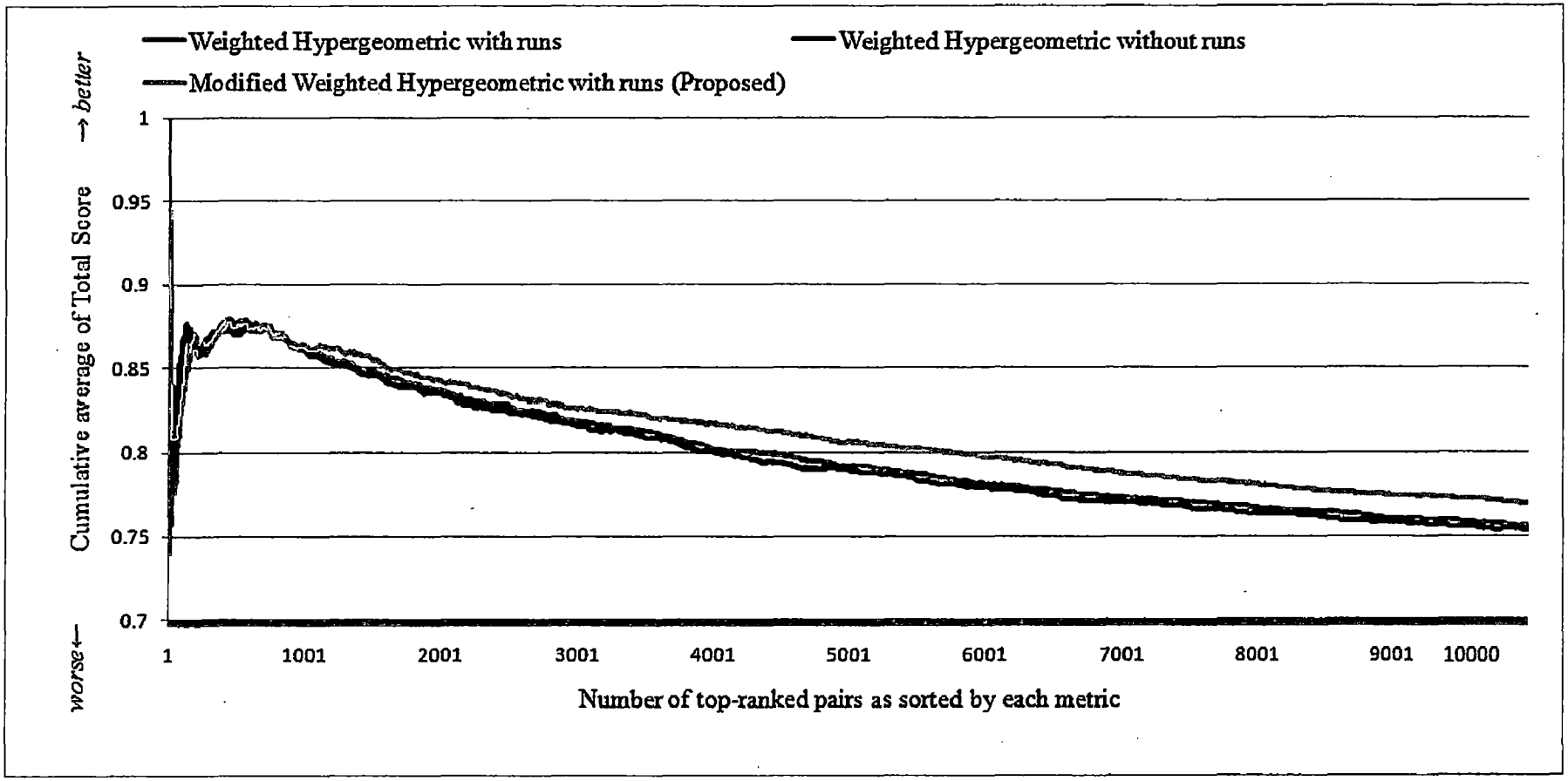


Figure 5.1 Pairwise Comparison of Phylogenetic Profiles

Each method assigns a p-value to every pair of genes (1 lakh pairs). Then gene pairs are sorted in ascending order by this p-value. Graph in Figure 5.1 is plotted as given x-axis value x, y is plotted as mean(total score) of first x gene pairs after sorting based on p-value. Where total score is the score which is obtained from the STRING database. The score is ranged from value 0 to 1. The greater the value the more the functional relatedness between the proteins. From the graph it shows that the pairs obtained from the propose method modified weighted hypergeometric probability with runs (green line) outperformed the other two methods weighted hypergeometric probability with runs (blue line) and weighted hypergeometric probability without runs (red line).

Table 5.1 Cumulative Average of Total Score for top 10,000 pairs

Method	**Cumulative Average of Total Score**
Modified Hypergeometric Probability with runs (Proposed)	0.76945
Hypergeometric Probability with runs	0.756
Hypergeometric Probability without runs	0.753

The cumulative average considering the 10,000 pairs for the proposed method is 0.76945, where as the values of weighted hypergeometric probability with runs is 0.756 and for weighted hypergeometric probability without runs is 0.753 as shown in Table 5.1. In the paper which proposed weighted hypergeometric probability with runs compared the results with gene ontologies and showed that it out performed weighted hypergeometric probability without runs with a big margin. If we see that margin in this case between those two methods is 0.003, though numerically low value since it is an average on 10,000 pairs it is good in this case. If we see our proposed method, it outperformed the best method existing which is weighted hypergeometric probability with runs by a value of 0.01345 which is a big margin in term of interaction score given by STRING database.

5.1.2 Network Degree Distribution

A network (an undirected graph with no multiple edges and no self-edges) is obtained from a computational method by ranking gene pairs by the p-values from that method and then collecting the top ranked 10,000 pairs. The nodes are the genes mentioned in the kept gene pairs and an edge is placed between two different genes if and only if the gene pair consisting of the two genes is among the kept gene pairs. The degree of a node is the number of edges incident with that node.

The Figure 5.2 shows two histograms (with a logarithmic scale for frequency) of node degree, one (blue) for the network from the weighted hypergeometric with runs computational method and the other (red) for the modified weighted hypergeometric with runs computational method.

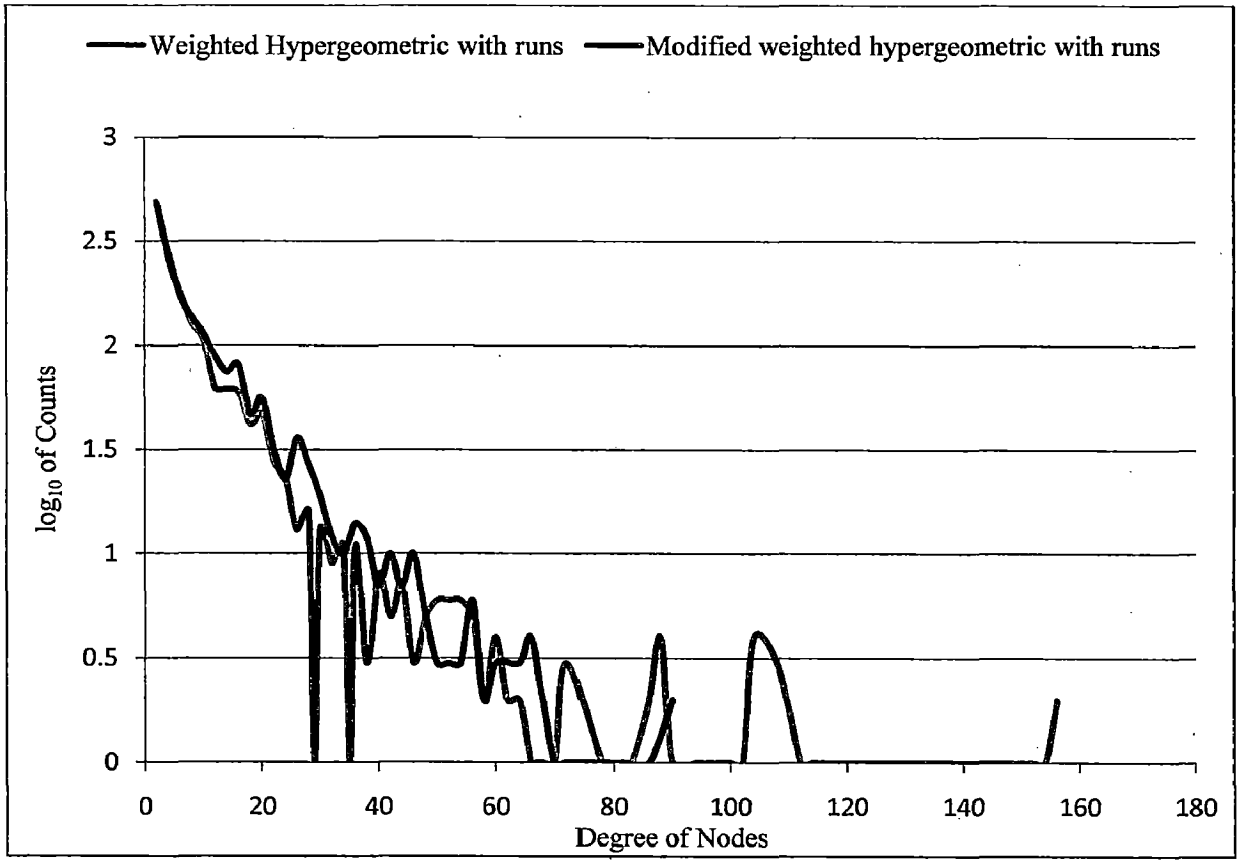


Figure 5.2 Network Degree Distribution

In Figure 5.2, we can see that proposed weighted hypergeometric with runs network (blue line) contains may nodes with 90 or more edges, while the modified weighted with runs (red line) has almost none. By seeing the above graph, we can say that the network

formed by proposed method is broken down into smaller clusters when compared to the pure weighted hypergeometric probability with runs. This is significant because large clusters are not very useful for functional studies since they bring together proteins with a broad range of functions. In contrast, small clusters can contain proteins with well-defines functional relationships.

5.1.3 Analysis with an Example

We considered the subunits of nitrate reductase. The Figure 5.3 shows all the interactions of the six subunits of nitrate reducatase which are narY, narH, narZ, narV, narJ and narG. These interactions are which are present in the STRING database.

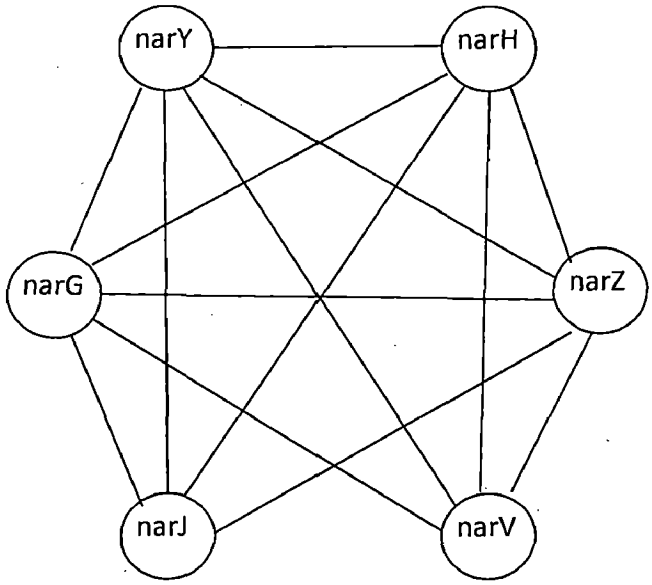


Figure 5.3 Functional Interactions between Proteins in Nitrate Reductase in the STRING Database.

The Figure 5.4 shows network containing all the interactions of the six subunits of nitrate reducatase which are observed using our proposed methodology modified weighted hypergeometric probability with runs. These proteins belong together as they are subunits of a protein complex that catalyzes the reduction of nitrate to ammonia. In the network generated using our methodology, the interactions missing are narG-narZ, narZ-narJ and narJ-narG and these links were of less score implies that less significant edges. The

network obtained is almost near to the existing true interactions which are found in the database.

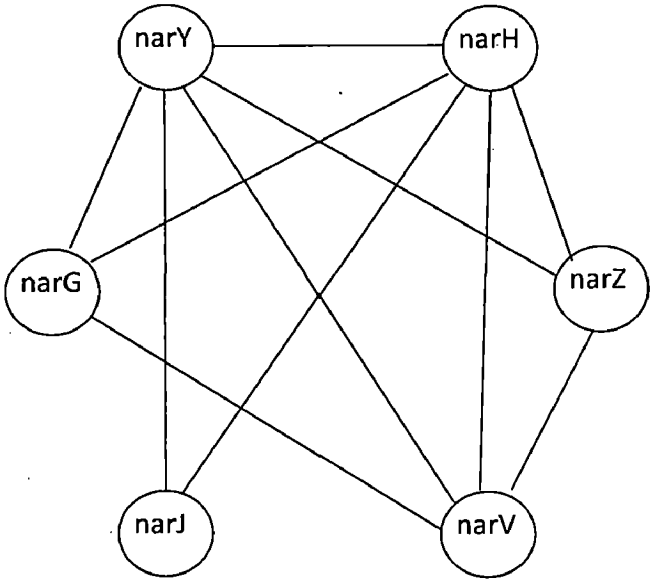


Figure 5.4 Functional Interactions between Proteins in Nitrate Reductase Using Proposed Methodology.

5.2 Results of Functional Catalogue Database Method

SVM is trained for each functional category to predict whether a gene should be assigned to it or not based on phylogenetic profiles. Using 3-fold cross validation repeated 10 times for each class, we compared the performance of proposed kernel function to SVM with the linear kernel, polynomial kernel [18] and tree kernel [17] through their Receiver Operating Characteristic (ROC) curves, i.e., the plot of true positives as a function of false positives.

5.2.1 ROC 50 Scores

Table 5.2 shows the functional categories of the level 1 in functional class tree and their ROC50 scores obtained by SVM using linear kernel, polynomial kernel, radial basis function (RBF) and tree kernel. It show that the performance of the radial basis kernel is similar to polynomial kernel is some functional classes. However for the class TRANSPORT FACILITATION both linear kernel and polynomial kernel has much higher ROC50 score and for class TRANSCRIPTION all the four kernels have almost

same ROC50 score and this tend to be larger and more general classes than other. Over all radial basis kernel outperformed the polynomial kernel linear kernels and tree kernel.

Table 5.2 ROC50 scores for the predictions of the level 1 classes in the functional class tree using 4 kernel functions.

Functional Class	Linear	Polynomial	Tree	RBF (Proposed Method)
METABOLISM	0.242	0.272	0.218	0.323
ENERGY	0.099	0.265	0.105	0.29
PROTEIN SYNTHESIS	0.061	0.274	0.186	0.288
CELLULAR ORGANIZATION	0.169	0.218	0.221	0.229
IONIC HOMEOSTASIS	0.047	0.179	0.105	0.217
TRANSPORT FACILITATION	0.318	0.381	0.273	0.19
CELLULAR TRANSPORT AND TRANSPORTMECHANISMS	0.072	0.109	0.054	0.147
CELL RESCUE, DEFENSE, CELL DEATH AND AGEING	0.054	0.113	0.049	0.141
TRANSCRIPTION	0.114	0.125	0.117	0.122
CELL GROWTH, CELL DIVISION AND DNA SYNTHESIS	0.059	0.08	0.06	0.086
PROTEIN DESTINATION	0.048	0.09	0.04	0.078
CELLULAR COMMUNICATION/SIGNAL TRANSDUCTION	0.057	0.122	0.079	0.062
CELLULAR BIOGENESIS	0.03	0.031	0.034	0.033

The ROC50 scores of the 50 largest protein functions in descending order sorted by size are plotted in Figure 5.5. The sum of the ROC50 scores over the 50 functions for the linear, polynomial, radial basis and tree kernel are 4.8, 8.0, 7.5 and 6.0 respectively.

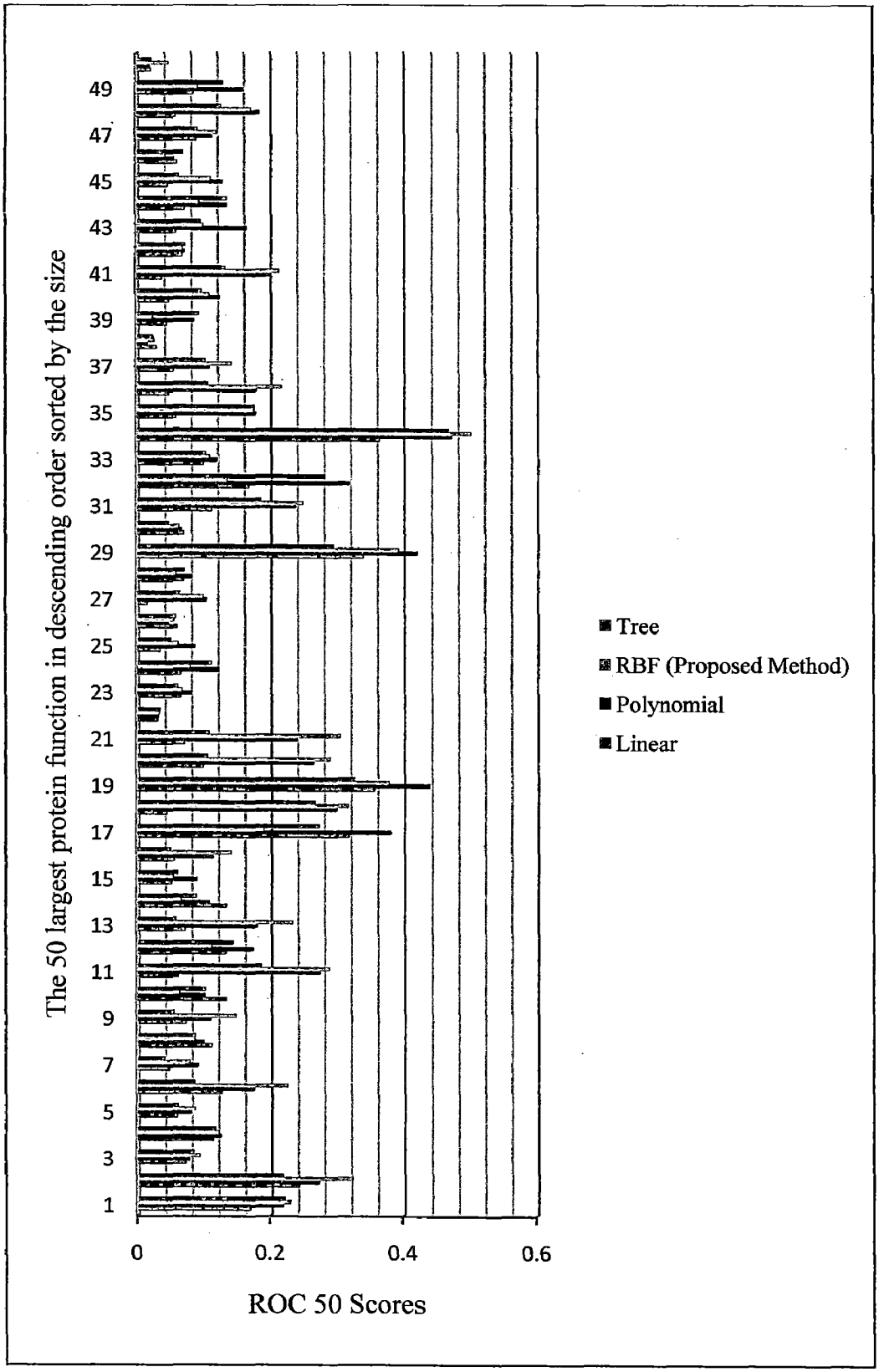


Figure 5.5 Comparison of 4 SVM Kernel types.

Even though radial basis has outperformed all three kernels, polynomial kernel has high sum. The reason for this is that the polynomial kernel has much higher values than other kernels in classes like amino-acid metabolism (0.437) and transport facilitation (0.381).

5.2.2 Class Wise Results

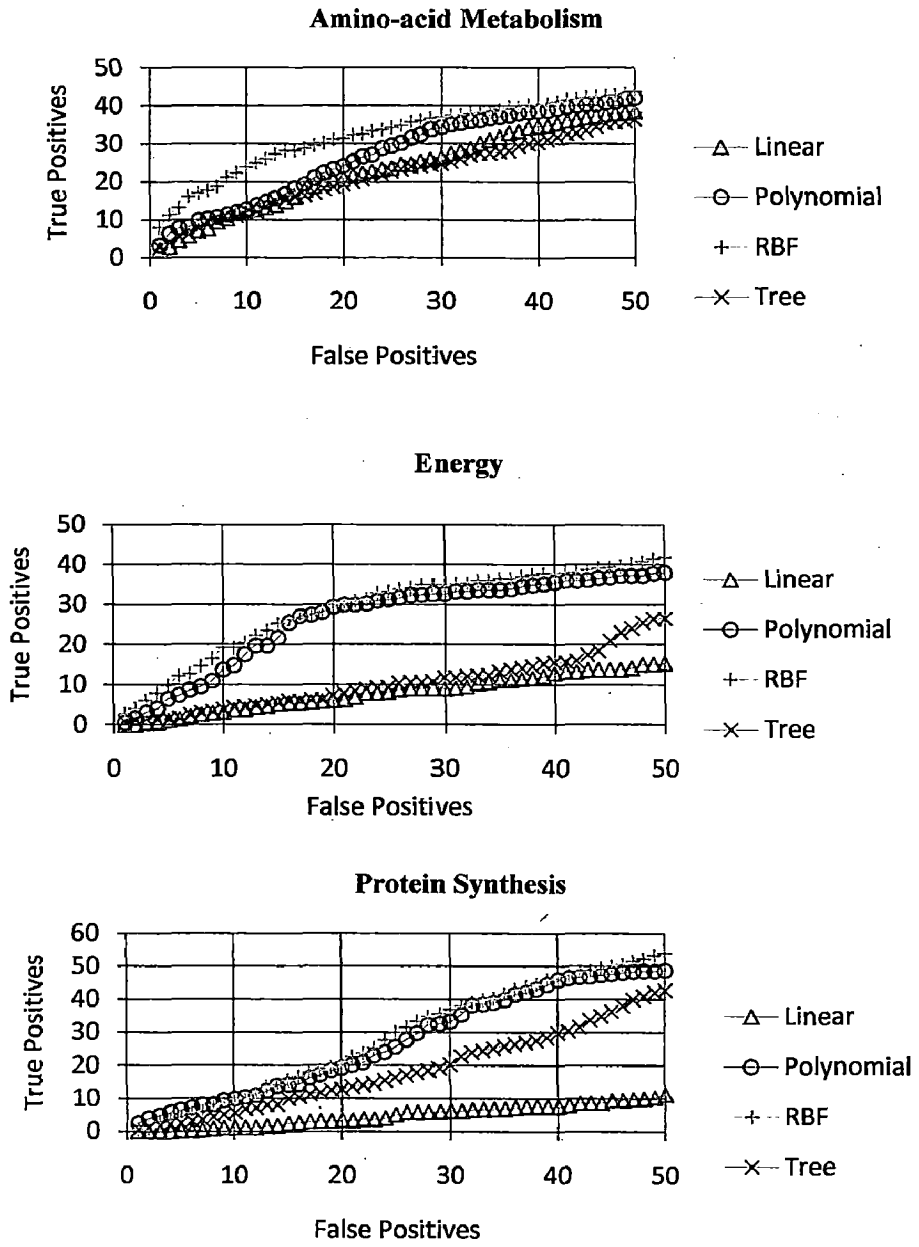


Figure 5.6 ROC Curves

Figures 5.6 shows the ROC curves up to 50 false positives corresponding to the top three classes with the highest radial basis function ROC score, in order to further study the differences in performance. These plots show that in several cases the RBF kernel significantly outperforms the other three kernels.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this work, we explored the possibility of predicting protein function using phylogenetic profiles. We proposed two solutions, first one using Functional Protein Association Network and second one using Functional Catalogue Database.

In the first method using Functional Protein Association Network, we used the probabilistic approach to incorporate the two important aspects of functional relatedness which are similarity measure and the number of runs the profiles span given the ordering of genomes. We tested the method using the 4195 phylogenetic profiles of Escherichia coli K12 generated using 305 genomes.

The following conclusions can be made from the results obtained using the proposed system and above mentioned data:

- The proposed method can yield good predictions based on number of reference genomes. Greater the number of genomes considered as reference genomes for profiles generation better will be the predictions.
- The cumulative average of STRINGS score considering the top ranked 10,000 pairs for the proposed method is 0.76945, where as the values of weighted hypergeometric probability with runs is 0.756 and for weighted hypergeometric probability without runs is 0.753.
- Our proposed method outperformed the best method existing which is weighted hypergeometric probability with runs by a value of 0.01345 which is a big margin in term of interaction score given by STRINGS database.
- The proposed method is not computationally expensive while considering phylogeny for identifying links
- This method is applicable to any target genome.

In the second method using Functional Catalogue Database, we classified phylogenetic profiles using supervised machine learning method, support vector machine classification along with radial basis function as kernel for identifying functionally linked proteins. We tested the algorithm for 2465 annotated genes from the yeast genome.

We compared the results with linear kernel, polynomial kernel and tree kernel. Polynomial kernel and RBF kernel together gave prediction more accurate than linear and tree kernel. Overall RBF kernel outperformed other three kernels.

Hence, this work can aid the understanding of protein functions for biomedical researchers and assist database curators in annotating protein functions and interactions efficiently, thus promoting the progress of genomics research.

6.2 Future Work

There is obviously significant room for improving the methods that we used for the prediction of protein function. The possible improvements in the future are listed as below:

- In the optimal leaf ordering step there are optimization steps involved in the implementation stage which is itself an area of research.
- For incorporating phylogeny we used concepts of runs, other optimal methods of incorporating phylogeny can be explored.
- Since the profiles are binary vectors, an entire field of data mining known as association analysis has been dedicated to this kind of data which is yet to be explored.
- Set of phylogenetic profiles can be treated as a binary matrix, which maps directly to the concept of market basket analysis.

In the future, there will be enormous need for the rapid annotations of protein functions and interactions for biomedical researchers to access the biomedical problems of human beings and to prescribe the drugs for their cure.

REFERENCES

[1] Pellegrini M, Marcotte E M, Thompson M J, Eisenberg D and Yeates T O, "Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles", Proceedings of the National Academy of Sciences, Vol. 96, No. 8, pp. 4285-4288, April 1999.

[2] "National Center for Biotechnology Information", [Online] Available: http://www.ncbi.nlm.nih.gov [Last Accessed 15 May 2009].

[3] Hunter L, "Chapter 1: Molecular Biology for Computer Scientists", In Artificial Intelligence and Molecular Biology. Ed: Hunter L.MIT Press, 1993. [Online] Available: www.biostat.wisc.edu/~craven/hunter.pdf [Last Accessed 15 May 2009].

[4] UniProtKB/TrEMBL Protein Database Release 40.3 Statistics, [Online] Available: http://www.ebi.ac.uk/uniprot/TrEMBLstats [Last Accessed 1 June 2009].

[5] David Lee, Oliver Redfern and Christine Orengo, "Predicting protein function from sequence and structure", Nature reviews molecular cell biology, Vol. 8, No. 12, pp. 995-1005, December 2007.

[6] "BioCreAtIve 2", 2006-2007, [Online] Available: http://biocreative.sourceforge.net/ [Last Accessed 15 May 2009].

[7] "Gene Ontology", [Online] Available: http://www.geneontology.org/ [Last Accessed 15 May 2009].

[8] Rost B, Liu J, Nair R, Wrzeszczynski KO and Ofran Y, "Automatic prediction of protein function", Cell Mol Life Sci, Vol. 60, No. 12, pp. 2637-50, December 2003.

[9] "The automated function prediction special interest group meeting", [Online] Available: http://compbio.iupui.edu/afp/2008 [Last Accessed 15 May 2009].

[10] D Szafron, P Lu, R Greiner, D Wishart, Z Lu, B Poulin, R Eisner, J Anvik and C Macdonell, "Proteome Analyst - Transparent High-throughput Protein

Annotation: Function, Localization and Custom Predictors", ICML Workshop - Bioinformatics, pp. 2–10, August 2003.

[11] T Hastie, R Tibshirani and J Friedman, "The Elements of Statistical Learning", Springer Series in Statistics, Springer, 2001.

[12] Kotsiantis S B, "Supervised Machine Learning: A Review of Classification Techniques", Informatica Journal, Vol. 31, No. 1, pp. 249-268, July 2007.

[13] Wu J, Kasif S and Delisi C, "Identification of functional links between genes using phylogenetic profiles", Bioinformatics, Vol. 19, No. 12, pp. 1524-1530, February 2003.

[14] Jiawen Han and Micheline Kamber, "DATA MINING CONCEPTS AND TECHNIQUES", Second Edition, Morgan Kaufmann Publishers, India, 2007.

[15] Kanehisa M, Goto S, Kawashima S, Okuno Y and Hattori M, "The KEGG resource for deciphering the genome" Nucleic Acids Research, Vol. 32, Database issue. D277–D280, 2004.

[16] Appala Raju Kotaru and R C Joshi, "Classification of Phylogenetic Profiles: Protein Function Prediction", International Journal of BioSciences and Technology (IJBST) ISSN: 0974-3987, 2009. (Under Review).

[17] Vert J P, "A tree kernel to analyse phylogenetic profiles", Bioinformatics, Vol. 18, No. 1, pp. S276-84, 2002.

[18] Kishore Narra and Li Liao, "Use of Extended Phylogenetic Profiles with E-Values and Support Vector Machines for Protein Family Classification", International Journal of Computer and Information Science, Vol. 6, No. 1, pp. 58-63, 2005.

[19] Barker D and Pagel M, "Predicting functional gene links from phylogenetic-statistical analyses of whole genomes", PLoS Comput Biol, Vol. 1, No. 1, pp. e3, June 2005.

[20] Zhou Y, Wang R, Li L, Xia X and Sun Z, "Inferring functional linkages between proteins from evolutionary scenarios", Journal Mol Biol, Vol. 16, No. 4, pp. 1150-1159, June 2006.

[21] Barker D, Meade A and Pagel M, "Constrained models of evolution lead to improved prediction of functional linkage from correlated gain and loss of genes", Bioinformatics, Vol. 23, No. 1, pp. 14-20, November 2007.

[22] Cokus S, Mizutani S and Pellegrini M, "An improved method for identifying functionally linked proteins using phylogenetic profiles", BMC Bioinformatics, Vol. 8, No. 4, pp. S7, May 2007.

[23] Sun J, Xu J, Liu Z, Liu Q, Zhao A, Shi T and Li Y, "Refined phylogenetic profiles method for predicting protein-protein interactions", Bioinformatics, Vol. 21, No. 16, pp. 3409-3415, June 2005.

[24] Loganantharaj R and Atwi M, "Towards validating the hypothesis of phylogenetic profiling", BMC Bioinformatics, Vol 8, No 7, pp. S25, November 2007.

[25] Snitkin E S, Gustafson A M, Mellor J, Wu J and DeLisi C, "Comparative assessment of performance and genome dependence among phylogenetic profiling methods", BMC Bioinformatics, Vol 7, No. 1, pp. 420, September 2006.

[26] Raja Jothi, Teresa M Przytycka and L Aravind, "Discovering functional linkages and uncharacterized cellular pathways using phylogenetic profile comparisons: A comprehensive assessment", BMC Bioinformatics, Vol 8, No. 1, pp. 173, May 2007.

[27] Xing-Ming Zhao, Yong Wang, Luonan Chen and Kazuyuki Aihara, "Gene function prediction using labeled and unlabeled data", BMC Bioinformatics, Vol 9, No. 1, pp. 57, January 2008.

[28] Yanan M A, Pingping Sun, Yuanping Sun, Ying Cui and Zhiqiang M A, "Research of Improved Phylogenetic Profiling Method", Second International Conference on Bioinformatics and Biomedical Engineering, pp. 233-236, May 2008.

[29] Altschul S F, Madden T L, Schffer A A, Zhang J, Zhang Z, Miller W and Lipman D J, "Gapped BLAST and PSI-BLAST : a new generation of protein database search programs", Nucleic Acids Research, Vol. 25, No. 17, pp. 3389-3402, September 1997.

[30] Fitz-Gibbon S T and House C H, "Whole genome-based phylogenetic analysis of free-living microorganisms", Nucleic Acids Res, Vol. 27, No. 21, pp. 4218-4222, November 1999.

[31] Bar-Joseph Z, Demaine E D, Gifford D K, Srebro N, Hamel A M and Jaakkola T S, "K-ary clustering with optimal leaf ordering for gene expression data", Bioinformatics, Vol. 19, No. 9, pp. 1070-1078, June 2003.

[32] Supplementary material for "Ab initio genotype–phenotype association reveals intrinsic modularity in genetic networks", [Online] Available: http://tavazoielab.princeton.edu/genphen, [last accessed 15 st April 2009].

[33] B. Snel, G. Lehmann, P. Bork and M. A. Huynen, "STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene", Nucleic Acids Res, Vol. 28, No. 18, pp. 3442-3444, September 2000.

[34] Christian von Mering, Martijn Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork and Berend Sne, "STRING: a database of predicted functional associations between proteins", Nucleic Acids Res, Vol. 3, No. 1, pp. 258-261, January 2003.

[35] Christian von Mering, Lars J. Jensen, Berend Snel, Sean D Hooper, Markus Krupp, Mathilde Foglierini, Nelly Jouffre, Martijn A. Huynen and Peer Bork, "STRING: known and predicted protein–protein associations, integrated and transferred across organisms", Nucleic Acids Res, Vol. 33, Database issue: D433-437, January 2009.

[36] Lars J Jensen, Michael Kuhn, Manuel Stark, Samuel Chaffron, Chris Creevey, Jean Muller, Tobias Doerks, Philippe Julien, Alexander Roth, Milan Simonovic, Peer Bork and Christian von Mering, "STRING 8—a global view on proteins and their functional interactions in 630 organisms", Nucleic Acids Res, Vol. 37, Database issue: D412–D416, January 2009.

[37] Mewes H W, Fridhman D, Guldener U, Mannhaupt G, Mayer K, Mokrejs M, Morgenstern B, Munsterkoetter M, Rudd S and Weil B, "MIPS: a database for genomes and proteins sequences", Nucleic Acids Research, Vol. 30, No. 1, pp. 31-34, January 2002.

[38] Vapnik VN, "The Nature of Statistical Learning Theory" Springer Verlag, New York, 1995.

[39] Joachims T, "Making large-Scale SVM Learning Practical", Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, pp. 169-184, 1999.

[40] Gribskov M and Robinso N, "Use of receiver operating characteristic (roc) analysis to evaluate sequence matching", Computers and Chemistry, Vol. 20, No. 1, pp. 25-33, 1996.

LIST OF PUBLICATIONS

[1] **Appala Raju Kotaru**, R C Joshi, "Classification of Phylogenetic Profiles for Protein Function Prediction: An SVM Approach", International Conference on Contemporary Computing – Bioinformatics, August 2009. (Accepted and will be published by Springer in Communications in Computer and Information Science ISSN: 1865-0929, http://www.jiit.ac.in/jiit/ic3/).

[2] **Appala Raju Kotaru**, R C Joshi, "Classification of Phylogenetic Profiles: Protein Function Prediction", International Journal of BioSciences and Technology (IJBST) ISSN: 0974-3987. (Under Review, http://www.ijbst.org/Home/papers-under-review).

[3] **Appala Raju Kotaru**, R C Joshi, "Protein Function Prediction using Phylogenetic profiles", ICASME-09, Goa, April 2009. (Accepted).

APPENDIX: Genomes List and their Ordering

A. List of all the reference genomes considered for construction of Phylogenetic profiles.

S No.	Reference Genomes
1	Bifidobacterium_longum
2	Corynebacterium_diphtheriae
3	Corynebacterium_efficiens_YS-314
4	Corynebacterium_glutamicum_ATCC_13032_Kitasato
5	Corynebacterium_glutamicum_ATCC_13032_Bielefeld
6	Corynebacterium_jeikeium_K411
7	Frankia_CcI3
8	Leifsonia_xyli_xyli_CTCB0
9	Mycobacterium_avium_paratuberculosis
10	Mycobacterium_bovis
11	Mycobacterium_leprae
12	Mycobacterium_tuberculosis_CDC1551
13	Mycobacterium_tuberculosis_H37Rv
14	Nocardia_farcinica_IFM10152
15	Propionibacterium_acnes_KPA171202
16	Streptomyces_avermitilis
17	Streptomyces_coelicolor
18	Symbiobacterium_thermophilum_IAM14863
19	Thermobifida_fusca_YX
20	Tropheryma_whipplei_TW08_27
21	Tropheryma_whipplei_Twist
22	Agrobacterium_tumefaciens_C58_UWash
23	Agrobacterium_tumefaciens_C58_Cereon
24	Anaplasma_marginale_St_Maries
25	Bartonella_henselae_Houston-1
26	Bartonella_quintana_Toulouse
27	Bradyrhizobium_japonicum
28	Brucella_abortus_9-941
29	Brucella_melitensis
30	Brucella_melitensis_biovar_Abortus
31	Brucella_suis_1330
32	Candidatus_Pelagibacter_ubique_HTCC1062
33	Caulobacter_crescentus
34	Ehrlichia_canis_Jake
35	Ehrlichia_ruminantium_Gardel

36	Ehrlichia_ruminantium_str._Welgevonden
37	Ehrlichia_ruminantium_Welgevonden
38	Erythrobacter_litoralis_HTCC2594
39	Gluconobacter_oxydans_621H
40	Magnetospirillum_magneticum_AMB-1
41	Mesorhizobium_loti
42	Nitrobacter_winogradskyi_Nb-255
43	Novosphingobium_aromaticivorans_DSM_12444
44	Rhizobium_etli_CFN_42
45	Rhodobacter_sphaeroides_2_4_1
46	Rhodopseudomonas_palustris_CGA009
47	Rhodopseudomonas_palustris_HaA2
48	Rhodospirillum_rubrum_ATCC_11170
49	Rickettsia_conorii
50	Rickettsia_felis_URRWXCal2
51	Rickettsia_prowazekii
52	Rickettsia_typhi_wilmington
53	Silicibacter_pomeroyi_DSS-3
54	Sinorhizobium_meliloti
55	Wolbachia_endosymbiont_of_Drosophila_melanogaster
56	Wolbachia_endosymbiont_of_Brugia_malayi_TRS
57	Zymomonas_mobilis_ZM4
58	Aquifex_aeolicus
59	Bacteroides_fragilis_NCTC_9434
60	Bacteroides_fragilis_YCH46
61	Bacteroides_thetaiotaomicron_VPI-5482
62	Chlorobium_chlorochromatii_CaD3
63	Chlorobium_tepidum_TLS
64	Pelodictyon_luteolum_DSM_273
65	Porphyromonas_gingivalis_W83
66	Salinibacter_ruber_DSM_13855
67	Azoarcus_sp_EbN1
68	Bordetella_bronchiseptica
69	Bordetella_parapertussis
70	Bordetella_pertussis
71	Burkholderia_mallei_ATCC_23344
72	Burkholderia_pseudomallei_1710b
73	Burkholderia_pseudomallei_K96243
74	Burkholderia_383

75	Burkholderia_thailandensis_E264
76	Chromobacterium_violaceum
77	Dechloromonas_aromatica_RCB
78	Neisseria_gonorrhoeae_FA_1090
79	Neisseria_meningitidis_MC58
80	Neisseria_meningitidis_Z2491
81	Nitrosomonas_europaea
82	Nitrosospira_multiformis_ATCC_25196
83	Ralstonia_eutropha_JMP134
84	Ralstonia_solanacearum
85	Thiobacillus_denitrificans_ATCC_25259
86	Parachlamydia_sp_UWE25
87	Chlamydia_muridarum
88	Chlamydia_trachomatis_A_HAR-13
89	Chlamydia_trachomatis
90	Chlamydophila_abortus_S26_3
91	Chlamydophila_caviae
92	Chlamydophila_pneumoniae_AR39
93	Chlamydophila_pneumoniae_CWL029
94	Chlamydophila_pneumoniae_J138
95	Chlamydophila_pneumoniae_TW_183
96	Dehalococcoides_ethenogenes_195
97	Dehalococcoides_CBDB1
98	Aeropyrum_pernix
99	Pyrobaculum_aerophilum
100	Sulfolobus_acidocaldarius_DSM_639
101	Sulfolobus_solfataricus
102	Sulfolobus_tokodaii
103	Anabaena_variabilis_ATCC_29413
104	Gloeobacter_violaceus
105	Nostoc_sp
106	Prochlorococcus_marinus_MIT_9312
107	Prochlorococcus_marinus_MIT9313
108	Prochlorococcus_marinus_NATL2A
109	Prochlorococcus_marinus_CCMP1375
110	Prochlorococcus_marinus_MED4
111	Synechococcus_elongatus_PCC_6301
112	Synechococcus_elongatus_PCC_7942
113	Synechococcus_CC9605

114	Synechococcus_CC9902
115	Cyanobacteria_bacterium_Yellowstone_B-Prime
116	Cyanobacteria_bacterium_Yellowstone_A-Prime
117	Synechococcus_sp_WH8102
118	Synechocystis_PCC6803
119	Thermosynechococcus_elongatus
120	Deinococcus_radiodurans
121	Thermus_thermophilus_HB27
122	Thermus_thermophilus_HB8
123	Bdellovibrio_bacteriovorus
124	Desulfotalea_psychrophila_LSv54
125	Desulfovibrio_desulfuricans_G20
126	Desulfovibrio_vulgaris_Hildenborough
127	Geobacter_metallireducens_GS-15
128	Geobacter_sulfurreducens
129	Pelobacter_carbinolicus
130	Campylobacter_jejuni_RM1221
131	Campylobacter_jejuni
132	Helicobacter_hepaticus
133	Helicobacter_pylori_26695
134	Helicobacter_pylori_J99
135	Thiomicrospira_denitrificans_ATCC_33889
136	Wolinella_succinogenes
137	Archaeoglobus_fulgidus
138	Haloarcula_marismortui_ATCC_43049
139	Halobacterium_sp
140	Methanococcus_jannaschii
141	Methanococcus_maripaludis_S2
142	Methanopyrus_kandleri
143	Methanosarcina_acetivorans
144	Methanosarcina_barkeri_fusaro
145	Methanosarcina_mazei
146	Methanosphaera_stadtmanae
147	Methanobacterium_thermoautotrophicum
148	Natronomonas_pharaonis
149	Picrophilus_torridus_DSM_9790
150	Pyrococcus_abyssi
151	Pyrococcus_furiosus
152	Pyrococcus_horikoshii

153	Thermococcus_kodakaraensis_KOD1
154	Thermoplasma_acidophilum
155	Thermoplasma_volcanium
156	Aster_yellows_witches-broom_phytoplasma_AYWB
157	Bacillus_anthracis_Ames_0581
158	Bacillus_anthracis_Ames
159	Bacillus_anthracis_str_Sterne
160	Bacillus_cereus_ATCC_10987
161	Bacillus_cereus_ATCC14579
162	Bacillus_cereus_ZK
163	Bacillus_clausii_KSM-K16
164	Bacillus_halodurans
165	Bacillus_licheniformis_ATCC_14580
166	Bacillus_licheniformis_DSM_13
167	Bacillus_subtilis
168	Bacillus_thuringiensis_konkukian
169	Carboxydothermus_hydrogenoformans_Z-2901
170	Clostridium_acetobutylicum
171	Clostridium_perfringens
172	Clostridium_tetani_E88
173	Enterococcus_faecalis_V583
174	Geobacillus_kaustophilus_HTA426
175	Lactobacillus_acidophilus_NCFM
176	Lactobacillus_johnsonii_NCC_533
177	Lactobacillus_plantarum
178	Lactobacillus_sakei_23K
179	Lactococcus_lactis
180	Listeria_innocua
181	Listeria_monocytogenes
182	Listeria_monocytogenes_4b_F2365
183	Mesoplasma_florum_L1
184	Moorella_thermoacetica_ATCC_39073
185	Mycoplasma_capricolum_ATCC_27343
186	Mycoplasma_gallisepticum
187	Mycoplasma_genitalium
188	Mycoplasma_hyopneumoniae_232
189	Mycoplasma_hyopneumoniae_7448
190	Mycoplasma_hyopneumoniae_J
191	Mycoplasma_mobile_163K

192	Mycoplasma_mycoides
193	Mycoplasma_penetrans
194	Mycoplasma_pneumoniae
195	Mycoplasma_pulmonis
196	Mycoplasma_synoviae_53
197	Oceanobacillus_iheyensis
198	Onion_yellows_phytoplasma
199	Staphylococcus_aureus_RF122
200	Staphylococcus_aureus_COL
201	Staphylococcus_aureus_aureus_MRSA252
202	Staphylococcus_aureus_aureus_MSSA476
203	Staphylococcus_aureus_MW2
204	Staphylococcus_aureus_Mu50
205	Staphylococcus_aureus_N315
206	Staphylococcus_aureus_NCTC_8325
207	Staphylococcus_aureus_USA300
208	Staphylococcus_epidermidis_ATCC_12228
209	Staphylococcus_epidermidis_RP62A
210	Staphylococcus_haemolyticus
211	Staphylococcus_saprophyticus
212	Streptococcus_agalactiae_2603
213	Streptococcus_agalactiae_A909
214	Streptococcus_agalactiae_NEM316
215	Streptococcus_mutans
216	Streptococcus_pneumoniae_R6
217	Streptococcus_pneumoniae_TIGR4
218	Streptococcus_pyogenes_M1_GAS
219	Streptococcus_pyogenes_MGAS10394
220	Streptococcus_pyogenes_MGAS315
221	Streptococcus_pyogenes_MGAS5005
222	Streptococcus_pyogenes_MGAS6180
223	Streptococcus_pyogenes_MGAS8232
224	Streptococcus_pyogenes_SSI-1
225	Streptococcus_thermophilus_CNRZ1066
226	Streptococcus_thermophilus_LMG_18311
227	Thermoanaerobacter_tengcongensis
228	Ureaplasma_urealyticum
229	Fusobacterium_nucleatum
230	Acinetobacter_sp_ADP1

231	Buchnera_sp
232	Buchnera_aphidicola
233	Buchnera_aphidicola_Sg
234	Candidatus_Blochmannia_floridanus
235	Candidatus_Blochmannia_pennsylvanicus_BPEN
236	Colwellia_psychrerythraea_34H
237	Coxiella_burnetii
238	Erwinia_carotovora_atroseptica_SCRI1043
239	Escherichia_coli_CFT073
240	Escherichia_coli_K12
241	Escherichia_coli_O157H7
242	Escherichia_coli_O157H7_EDL933
243	Francisella_tularensis_tularensis
244	Haemophilus_ducreyi_35000HP
245	Haemophilus_influenzae_86_028NP
246	Haemophilus_influenzae
247	Hahella_chejuensis_KCTC_2396
248	Idiomarina_loihiensis_L2TR
249	Legionella_pneumophila_Lens
250	Legionella_pneumophila_Paris
251	Legionella_pneumophila_Philadelphia_1
252	Mannheimia_succiniciproducens_MBEL55E
253	Methylococcus_capsulatus_Bath
254	Nitrosococcus_oceani_ATCC_19707
255	Pasteurella_multocida
256	Photobacterium_profundum_SS9
257	Photorhabdus_luminescens
258	Pseudoalteromonas_haloplanktis_TAC125
259	Pseudomonas_aeruginosa
260	Pseudomonas_fluorescens_Pf-5
261	Pseudomonas_fluorescens_PfO-1
262	Pseudomonas_putida_KT2440
263	Pseudomonas_syringae_phaseolicola_1448A
264	Pseudomonas_syringae_pv_B728a
265	Pseudomonas_syringae
266	Psychrobacter_arcticum_273-4
267	Salmonella_enterica_Choleraesuis
268	Salmonella_enterica_Paratypi_ATCC_9150
269	Salmonella_typhi_Ty2

270	Salmonella_typhi
271	Salmonella_typhimurium_LT2
272	Shewanella_oneidensis
273	Shigella_boydii_Sb227
274	Shigella_dysenteriae
275	Shigella_flexneri_2a_2457T
276	Shigella_flexneri_2a
277	Shigella_sonnei_Ss046
278	Sodalis_glossinidius_morsitans
279	Thiomicrospira_crunogena_XCL-2
280	Vibrio_cholerae
281	Vibrio_fischeri_ES114
282	Vibrio_parahaemolyticus
283	Vibrio_vulnificus_CMCP6
284	Vibrio_vulnificus_YJ016
285	Wigglesworthia_brevipalpis
286	Xanthomonas_citri
287	Xanthomonas_campestris_8004
288	Xanthomonas_campestris
289	Xanthomonas_campestris_vesicatoria_85-10
290	Xanthomonas_oryzae_KACC10331
291	Xylella_fastidiosa
292	Xylella_fastidiosa_Temecula1
293	Yersinia_pestis_CO92
294	Yersinia_pestis_KIM
295	Yersinia_pestis_biovar_Mediaevails
296	Yersinia_pseudotuberculosis_IP32953
297	Nanoarchaeum_equitans
298	Pirellula_sp
299	Borrelia_burgdorferi
300	Borrelia_garinii_PBi
301	Leptospira_interrogans_serovar_Copenhageni
302	Leptospira_interrogans_serovar_Lai
303	Treponema_denticola_ATCC_35405
304	Treponema_pallidum
305	Thermotoga_maritima

B. Order of the reference genomes considered for construction of Phylogenetic profiles after running optimal leaf ordering algorithm.

275, 75, 23, 89, 103, 201, 282, 238, 44, 39, 130, 207, 93, 21, 69, 199, 295, 119, 136, 11, 8, 303, 254, 15, 105, 193, 76, 142, 64, 30, 57, 151, 195, 14, 160, 104, 153, 92, 294, 45, 107, 31, 258, 242, 25, 47, 68, 216, 91, 230, 219, 110, 13, 205, 86, 149, 95, 211, 108, 185, 115, 150, 225, 222, 220, 157, 228, 56, 112, 284, 276, 50, 26, 137, 236, 233, 111, 63, 138, 22, 196, 241, 131, 48, 132, 290, 251, 100, 79, 96, 271, 292, 19, 203, 162, 246, 74, 180, 248, 53, 192, 33, 41, 213, 217, 302, 152, 170, 123, 232, 281, 260, 147, 188, 259, 255, 114, 208, 268, 197, 223, 27, 101, 189, 257, 293, 144, 215, 204, 253, 24, 182, 183, 124, 280, 4, 7, 283, 60, 73, 70, 126, 186, 181, 300, 113, 267, 67, 87, 161, 289, 117, 28, 165, 148, 5, 55, 72, 234, 231, 134, 187, 298, 118, 169, 16, 194, 154, 40, 191, 38, 200, 209, 299, 190, 264, 279, 143, 198, 173, 140, 58, 171, 61, 224, 287, 62, 297, 291, 9, 36, 52, 202, 82, 42, 125, 167, 17, 237, 277, 116, 285, 77, 81, 128, 177, 159, 71, 54, 166, 304, 133, 3, 252, 163, 244, 206, 227, 129, 229, 99, 155, 214, 226, 262, 102, 305, 240, 178, 249, 245, 139, 10, 35, 20, 78, 278, 32, 286, 90, 235, 256, 184, 80, 274, 18, 127, 43, 288, 141, 6, 88, 65, 34, 266, 212, 270, 164, 247, 168, 210, 2, 29, 83, 66, 49, 46, 250, 179, 156, 296, 37, 172, 261, 85, 12, 243, 122, 221, 135, 146, 175, 263, 176, 109, 121, 97, 94, 84, 272, 273, 1, 120, 218, 98, 265, 145, 106, 59, 239, 301, 174, 158, 51, 269.

Narra et al. [18] used the extended real-valued profiles to incorporate phylogeny which did not basically compare the trees.

Barker et al. [21] reconstructed phylogenetic tree to identify proteins that appear to co-evolve is more complex and computationally expensive.

Cokus et al. [22] used runs for considering phylogeny but while calculating runs probability it used similarity rather than using runs of the pairs.

From all above gaps found in the earlier work done by many of the authors, following is the summary of some of the properties which are crucially required in a proposed scheme.

1. Co-evolution should be considered.
2. The background phylogeny of genomes in the profile should also be considered.
3. It should scale for the rapidly increasing number of reference genomes.
4. It should not be computationally expensive while considering the tree in the calculation of functional relatedness.
5. Solution should be applicable to any target genome.